

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

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



«30» мая 2025

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ **Б1.О.14«Методы программирования»**

Направление подготовки 02.03.03 Математическое обеспечение и
администрирование информационных систем

Направленность (профиль) Технологии разработки программных систем

Форма обучения очная

Квалификация бакалавр

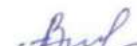
Краснодар 2025

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

Программу составил(и):

Харченко Анна Владимировна, доцент, канд. пед наук

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



подпись

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

Заведующий кафедрой (разработчика)

В. В. Подколзин



подпись

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

Заведующий кафедрой (выпускающей)

В. В. Подколзин



подпись

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

Председатель УМК факультета

А. В. Коваленко



подпись

Рецензенты:

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

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

1 Цели и задачи изучения дисциплины (модуля)

1.1 Цель освоения дисциплины

Целью курса является изучение фундаментальных структур данных (линейных, двунаправленных и кольцевых списков, двоичных деревьев, графов) и методов их обработки, а также приобретение практических навыков их реализации на языке программирования. Дисциплина формирует основы для применения современных средств вычислительной техники и программных технологий при решении задач в естественнонаучных, общепрофессиональных и специальных дисциплинах. Освоение теоретического материала и практическое применение структур данных, алгоритмов, файловых систем и контейнеров на базе современных ПК подготавливает студентов к использованию информационных систем и языков программирования в профессиональной деятельности.

1.2 Задачи дисциплины

Основные задачи включают освоение принципов организации и обработки связанных списков, включая линейные, двунаправленные и кольцевые структуры; изучение алгоритмов работы с двоичными деревьями (построение, обход, балансировка); знакомство с методами представления графов и базовыми алгоритмами их обработки (поиск в глубину, в ширину, кратчайшие пути). Важным аспектом является практическое применение структур данных при работе с файлами (чтение, запись, сериализация) и контейнерами, а также развитие навыков оптимизации кода и отладки программ.

1.3 Место дисциплины (модуля) в структуре образовательной программы

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

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

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций:

УК-1 **Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач**

ИД-1.УК-1 **Осуществляет поиск необходимой информации, опираясь на результаты анализа поставленной задачи**

Знать *Современные объектно-ориентированные языки программирования
Цели и задачи проводимых исследований и разработок
Методы и средства планирования и организации исследований и разработок*

Уметь *Проводить анализ исполнения требований
Верифицировать структуру программного кода
Применять методы анализа научно-технической информации*

Владеть *Анализ возможностей реализации требований к программному обеспечению
Разработка, изменение и согласование архитектуры программного обеспечения с системным аналитиком и архитектором программного обеспечения*

Верификация структуры программного кода ИС относительно архитектуры ИС и требований заказчика к ИС
Проведение маркетинговых исследований научно-технической информации
Сбор, обработка, анализ и обобщение передового отечественного и международного опыта в соответствующей области исследований
Подготовка предложений для составления планов и методических программ исследований и разработок, практических рекомендаций по исполнению их результатов
Деятельность, направленная на решение задач аналитического характера, предполагающих выбор и многообразие актуальных способов решения задач

ИД-2.УК-1 Выбирает оптимальный вариант решения задачи, аргументируя свой выбор

Знать *Современные объектно-ориентированные языки программирования*
Цели и задачи проводимых исследований и разработок
Методы и средства планирования и организации исследований и разработок

Уметь *Вырабатывать варианты реализации требований*
Проводить оценку и обоснование рекомендуемых решений
Верифицировать структуру программного кода
Применять методы анализа научно-технической информации

Владеть *Оценка и согласование сроков выполнения поставленных задач*
Разработка, изменение и согласование архитектуры программного обеспечения с системным аналитиком и архитектором программного обеспечения
Оценка и согласование сроков выполнения поставленных задач
Верификация структуры программного кода ИС относительно архитектуры ИС и требований заказчика к ИС
Проведение маркетинговых исследований научно-технической информации
Сбор, обработка, анализ и обобщение передового отечественного и международного опыта в соответствующей области исследований
Подготовка предложений для составления планов и методических программ исследований и разработок, практических рекомендаций по исполнению их результатов
Деятельность, направленная на решение задач аналитического характера, предполагающих выбор и многообразие актуальных способов решения задач

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

ИД-3.УК-2 Использует принципы проектной методологии для решения профессиональных задач

Знать *Методологии разработки программного обеспечения и технологии программирования*
Инструменты и методы верификации структуры программного кода
Цели и задачи проводимых исследований и разработок
Методы анализа и обобщения отечественного и международного опыта в соответствующей области исследований

Методы и средства планирования и организации исследований и разработок

Уметь *Вырабатывать варианты реализации требований
Проводить оценку и обоснование рекомендуемых решений
Применять нормативную документацию в соответствующей области знаний
Применять методы анализа научно-технической информации*

Владеть *Анализ возможностей реализации требований к программному обеспечению
Оценка времени и трудоемкости реализации требований к программному обеспечению
Согласование требований к программному обеспечению с заинтересованными сторонами
Сбор, обработка, анализ и обобщение передового отечественного и международного опыта в соответствующей области исследований
Деятельность, направленная на решение задач аналитического характера, предполагающих выбор и многообразие актуальных способов решения задач*

ИД-4.УК-2 ***Выбирает оптимальный способ решения задач, имеющихся ресурсов и ограничений, оценки рисков на основе проектного инструментария***

Знать *Методологии разработки программного обеспечения и технологии программирования
Инструменты и методы верификации структуры программного кода
Цели и задачи проводимых исследований и разработок
Методы анализа и обобщения отечественного и международного опыта в соответствующей области исследований
Методы и средства планирования и организации исследований и разработок*

Уметь *Вырабатывать варианты реализации требований
Применять нормативную документацию в соответствующей области знаний
Применять методы анализа научно-технической информации*

Владеть *Оценка времени и трудоемкости реализации требований к программному обеспечению
Согласование требований к программному обеспечению с заинтересованными сторонами
Оценка и согласование сроков выполнения поставленных задач
Оценка и согласование сроков выполнения поставленных задач
Сбор, обработка, анализ и обобщение передового отечественного и международного опыта в соответствующей области исследований
Деятельность, направленная на решение задач аналитического характера, предполагающих выбор и многообразие актуальных способов решения задач*

ОПК-2 **Способен применять современный математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества**

программных продуктов и программных комплексов в различных областях человеческой деятельности

ИД-1.ОПК-2 Способен применять системный подход к анализу предметной (проблемной) области, выявлению требований к ИС

Знать Методологии разработки программного обеспечения и технологии программирования
Цели и задачи проводимых исследований и разработок
Методы проведения экспериментов и наблюдений, обобщения и обработки информации

Уметь Проводить анализ исполнения требований
Использовать существующие типовые решения и шаблоны проектирования программного обеспечения

Владеть Анализ возможностей реализации требований к программному обеспечению
Согласование требований к программному обеспечению с заинтересованными сторонами
Разработка, изменение и согласование архитектуры программного обеспечения с системным аналитиком и архитектором программного обеспечения
Проектирование структур данных
Разработка структуры программного кода ИС
Сбор, обработка, анализ и обобщение результатов экспериментов и исследований в соответствующей области знаний
Деятельность, направленная на решение задач аналитического характера, предполагающих выбор и многообразие актуальных способов решения задач

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

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

Уметь Вырабатывать варианты реализации требований
Использовать существующие типовые решения и шаблоны проектирования программного обеспечения
Верифицировать структуру программного кода

Владеть Разработка, изменение и согласование архитектуры программного обеспечения с системным аналитиком и архитектором программного обеспечения
 Разработка структуры программного кода ИС
 Верификация структуры программного кода ИС относительно архитектуры ИС и требований заказчика к ИС
 Сбор, обработка, анализ и обобщение результатов экспериментов и исследований в соответствующей области знаний
 Деятельность, направленная на решение задач аналитического характера, предполагающих выбор и многообразие актуальных способов решения задач

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 5 зач. ед. (180 часов), их распределение по видам работ представлено в таблице

Вид учебной работы		Всего часов	Семестры (часы)					
			2					
Контактная работа, в том числе:		124,5	124,5					
Аудиторные занятия (всего):		118	118					
Занятия лекционного типа		50	50					
Лабораторные занятия		68	68					
Занятия семинарского типа (семинары, практические занятия)								
Иная контактная работа:		6,5	6,5					
Контроль самостоятельной работы (КСР)		6	6					
Промежуточная аттестация (ИКР)		0,5	0,5					
Самостоятельная работа, в том числе:		20	20					
Проработка учебного (теоретического) материала		6	6					
Выполнение индивидуальных заданий		10	10					
Подготовка к текущему контролю		4	4					
Контроль:		35,5	35,5					
Подготовка к экзамену		35,5	35,5					
Общая трудоемкость	час.	180	180					
	в том числе контактная работа	124,5	124,5					
	зач. ед	5	5					

2.2 Структура дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.
 Разделы (темы) дисциплины, изучаемые в 2 семестре

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	СРС
1	2	3	4	5	6	7
1.	Линейные динамические информационные структуры	18	8		8	2
2.	Линейные двунаправленные связные списки	26	10		14	2
3.	Кольцевые списки	10	4		4	2
4.	Двоичные деревья	20	6		12	2
5.	Файлы	14	4		6	4
6.	Контейнеры	26	10		12	4
7.	Обработка графов	24	8		12	4
ИТОГО по разделам дисциплины		138	50		68	20
Контроль самостоятельной работы (КСР)		6				
Промежуточная аттестация (ИКР)		0,5				
Подготовка к текущему контролю		35,5				
Общая трудоемкость по дисциплине		180				

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

2.3 Содержание разделов (тем) дисциплины

2.3.1 Занятия лекционного типа

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
1.	Линейные динамические информационные структуры	Информационные структуры. Линейные динамические информационные структуры. Основные определения примеры	К, РЗ
2.	Линейные двунаправленные связные списки	Стек. Очередь. Дек. Логическая организация. Основные операции. Абстрактные структуры	К, РЗ
3.	Линейные двунаправленные связные списки	Моделирование стека средствами языка С++. Набор основных операций	К, РЗ
4.	Линейные двунаправленные связные списки	Моделирование очереди средствами языка С++. Набор основных операций	К, РЗ
5.	Линейные двунаправленные связные списки	Связное распределение памяти. Однонаправленные списки. Построение списка. Операция вставки и удаления элемента. Двунаправленные списки. Построение, операция удаления элемента. Построение, операция вставки элемента. Рекурсивная обработка списков.	К, РЗ
6.	Кольцевые списки	Кольцевые списки. Построение, Обработка кольцевых списков.	К, РЗ

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
7.	Двоичные деревья	Деревья. Основные понятия. Дерево двоичного поиска (структура и построение). Обходы ДДП.	К, РЗ
8.	Файлы	Ввод-вывод. Система управления вводом-выводом. Блок FCB. Открытие и закрытие файлов. Буферизация ввода вывода	К, РЗ
9.	Файлы	Общие операции над файлами. Обработка текстовых, числовых бинарных файлов.	К, РЗ
10.	Обработка графов	Ориентированные графы. Основные определения. Представления ориентированных графов. АД для ориентированных графов. Основные операторы.	К, РЗ
11.	Обработка графов	Задача нахождения кратчайшего пути. Алгоритм Дейкстры. Реализация на C++.	К, РЗ
12.	Обработка графов	Нахождение кратчайших путей между парами вершин. Алгоритм Флойда. Реализация на C++.	К, РЗ
13.	Обработка графов	Представление неориентированных графов. Основные деревья минимальной стоимости. Основные понятия. Структура дерева	К, РЗ
14.	Обработка графов	Алгоритм Прима Реализация на C++	К, РЗ
15.	Обработка графов	Алгоритм Краскала. Реализация на C++	К, РЗ
16.	Контейнеры	Контейнерный класс vector. Основные методы, примеры использования.	К, РЗ
17.	Контейнеры	Контейнерный класс list. Основные методы, примеры использования.	К, РЗ
18.	Контейнеры	Контейнерный класс deque, queue. Основные методы, примеры использования.	К, РЗ
19.	Контейнеры	Ассоциативный контейнеры set. Основные методы, примеры использования.	К, РЗ
20.	Контейнеры	Ассоциативный контейнеры map. Основные методы, примеры использования.	К, РЗ

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

2.3.2 Занятия семинарского типа

Не предусмотрены

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

2.3.3 Лабораторные занятия

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
1.	Линейные динамические информационные структуры	Однонаправленные списки. Построение списка. Операция вставки	РЗ
2.	Линейные динамические информационные структуры	Однонаправленные списки. Построение списка. Операция удаления. элемента.	РЗ
3.	Линейные динамические информационные структуры	Рекурсивная обработка списков.	РЗ
4.	Линейные динамические информационные структуры	Решение задач с использованием очереди и стека	РЗ
5.	Линейные двунаправленные связные списки	Двунаправленные списки. Построение,	РЗ
6.	Линейные двунаправленные связные списки	Двунаправленные списки. Обработка в прямом и обратном порядке	РЗ
7.	Линейные двунаправленные связные списки	Двунаправленные списки. Операция вставки элемента	РЗ
8.	Линейные двунаправленные связные списки	Двунаправленные списки. Операция удаления элемента.	РЗ
9.	Линейные двунаправленные связные списки	Использование нескольких структур данных (массив, матрица, список) при решении задач	РЗ
10.	Линейные двунаправленные связные списки	Использование нескольких структур данных (стек, очередь, список) при решении задач	РЗ
11.	Линейные двунаправленные связные списки	Рекурсивная обработка двунаправленных списков	РЗ
12.	Кольцевые списки	Кольцевые списки. Построение	РЗ
13.	Кольцевые списки	Кольцевые списки. Обработка элементов	РЗ
14.	Двоичные деревья	Построение дерева двоичного поиска	РЗ
15.	Двоичные деревья	Обходы дерева двоичного поиска	РЗ
16.	Двоичные деревья	Обработка элементов дерева. Их свойства	РЗ
17.	Двоичные деревья	Уровни в деревьях	РЗ
18.	Двоичные деревья	Вставка элемента в дерево	РЗ
19.	Двоичные деревья	Удаление элемента из дерева	РЗ
20.	Файлы	Обработка целочисленных файлов	РЗ
21.	Файлы	Обработка текстовых файлов	РЗ
22.	Файлы	Обработка файлов разных форматов	РЗ
23.	Контейнеры	Обработка данных с помощью контейнера vector	РЗ
24.	Контейнеры	Обработка данных с помощью контейнера list	РЗ
25.	Контейнеры	Обработка данных с помощью контейнера deque	РЗ
26.	Контейнеры	Обработка данных с помощью контейнера queue	РЗ

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
27.	Контейнеры	Обработка данных с помощью контейнера set	РЗ
28.	Контейнеры	Обработка данных с помощью контейнера map	РЗ
29.	Обработка графов	Построение графа	РЗ
30.	Обработка графов	Обход в ширину	РЗ
31.	Обработка графов	Обход в глубине	РЗ
32.	Обработка графов	Алгоритм Флойда	РЗ
33.	Обработка графов	Алгоритм Дейкстры	РЗ
34.	Обработка графов	Построение дерева двоичного поиска	РЗ

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

2.3.4 Примерная тематика курсовых работ (проектов)

Не предусмотрены

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

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1	Изучение теоретического материала	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой информационных технологий, протокол №1 от 30.08.2019
2	Решение задач	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой информационных технологий, протокол №1 от 30.08.2019

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

Для лиц с нарушениями зрения:

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

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

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

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. Образовательные технологии

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

При обучении используются следующие образовательные технологии:

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

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

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

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

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

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

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

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

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

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

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

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

Основные виды интерактивных образовательных технологий включают в себя:

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

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

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

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

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

Семестр	Вид занятия	Используемые интерактивные образовательные технологии	количество интерактивных часов
	ЛР	Практические занятия в режимах взаимодействия «преподаватель – студент» и «студент – студент»	24
Итого			24

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

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

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

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

Для лиц с нарушениями зрения:

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

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

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

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

4. Оценочные и методические материалы

4.1 Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

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

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

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

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

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

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

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

Для лиц с нарушениями зрения:

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

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

Структура оценочных средств для текущей и промежуточной аттестации

№ п/п	Контролируемые разделы (темы) дисциплины*	Код контролируемой компетенции (или ее части)	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
1	Линейные динамические информационные структуры	ИД-1.УК-1 ИД-2.УК-1 ИД-3.УК-2 ИД-4.УК-2 ИД-1.ОПК-2 ИД-3.ОПК-2	Контрольные задания 1-4 Задания для самостоятельного решения 1-10	Вопрос на экзамене 1-3
2	Линейные двунаправленные связные списки	ИД-1.УК-1 ИД-2.УК-1 ИД-3.УК-2 ИД-4.УК-2 ИД-1.ОПК-2 ИД-3.ОПК-2	Контрольные задания 5-7 Задания для самостоятельного решения 11-17	Вопрос на экзамене 4-17
3	Кольцевые списки	ИД-1.УК-1 ИД-2.УК-1 ИД-3.УК-2 ИД-4.УК-2 ИД-1.ОПК-2 ИД-3.ОПК-2	Контрольные задания 8-9 Задания для самостоятельного решения 18-22	Вопрос на экзамене 2,4,11

4	Двоичные деревья	ИД-1.УК-1 ИД-2.УК-1 ИД-3.УК-2 ИД-4.УК-2 ИД-1.ОПК-2 ИД-3.ОПК-2	Контрольные задания 10-12 Задания для самостоятельного решения 23-33	Вопрос на экзамене 18-21
5	Файлы	ИД-1.УК-1 ИД-2.УК-1 ИД-3.УК-2 ИД-4.УК-2 ИД-1.ОПК-2 ИД-3.ОПК-2	Контрольные задания 13-14 Задания для самостоятельного решения 34-38	Вопрос на экзамене 22-24
6	Контейнеры	ИД-1.УК-1 ИД-2.УК-1 ИД-3.УК-2 ИД-4.УК-2 ИД-1.ОПК-2 ИД-3.ОПК-2	Контрольные задания 15-18 Задания для самостоятельного решения 39-52	Вопрос на экзамене 25-35
7	Обработка графов	ИД-1.УК-1 ИД-2.УК-1 ИД-3.УК-2 ИД-4.УК-2 ИД-1.ОПК-2 ИД-3.ОПК-2	Контрольные задания 19-20 Задания для самостоятельного решения 53-57	Вопрос на экзамене 36-42

Показатели, критерии и шкала оценки сформированных компетенций

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

УК-1 **Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач**

ИД-1.УК-1 **Осуществляет поиск необходимой информации, опираясь на результаты анализа поставленной задачи**

Знать Современные методы программирования на C++ (ООП, STL, шаблоны).
Принципы работы и реализации:
Линейных динамических структур (односвязные, двусвязные, кольцевые списки).
Деревьев (бинарные, AVL, красно-черные).
Графов (алгоритмы обхода, поиска кратчайшего пути).
Контейнеров STL (vector, list, map, set и др.).
Методы работы с файлами (текстовые, бинарные, сериализация).

Уметь Анализировать требования к структурам данных и выбирать оптимальную реализацию.

Проверять корректность реализации алгоритмов (верификация кода).
Применять методы анализа эффективности алгоритмов (O-нотация).
Владеть Разработкой и оптимизацией структур данных на C++.
Сравнительным анализом различных реализаций (например, std::list vs рукописный список).
Обработкой данных из файлов и их структурированным хранением.
Применением STL-контейнеров и алгоритмов для решения задач.

ИД-2.УК-1 **Выбирает оптимальный вариант решения задачи, аргументируя свой выбор**

Знать Преимущества и недостатки различных структур данных (массивы vs списки, хеш-таблицы vs деревья).

	Особенности работы с памятью в C++ (указатели, аллокация, кэш-локальность).
	Критерии выбора алгоритмов (временная и пространственная сложность).
Уметь	Сравнивать различные подходы к реализации (например, итеративный vs рекурсивный обход дерева).
	Обосновывать выбор контейнера STL для конкретной задачи.
	Анализировать производительность кода с помощью профилирования.
Владеть	Оптимизацией структур данных под конкретные задачи (кеширование, пулы памяти).
	Реализацией высокоэффективных алгоритмов на графах и деревьях.
	Использованием продвинутых возможностей STL (аллокаторы, пользовательские компараторы).
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
ИД-3.УК-2	Использует принципы проектной методологии для решения профессиональных задач
Знать	Алгоритмы и структуры данных, изучаемые в курсе.
	Особенности реализации в C++ (управление памятью, указатели, шаблоны).
	Критерии выбора структур (время доступа, сложность операций).
Уметь	Выбирать между разными структурами (например, массив vs список).
	Обосновывать выбор алгоритма (например, DFS vs BFS для графов).
	Анализировать производительность кода (профилирование).
Владеть	Реализацией динамических структур (списки, деревья) на C++.
	Применением STL (std::list, std::map, std::unordered_set).
	Оптимизацией работы с памятью (умные указатели, move-семантика).
ИД-4.УК-2	Выбирает оптимальный способ решения задач, имеющихся ресурсов и ограничений, оценки рисков на основе проектного инструментария
Знать	Методы работы с файлами в C++ (fstream, сериализация).
	Алгоритмы обработки графов (Дейкстра, Флойд, топологическая сортировка).
	Особенности контейнеров STL (выбор между vector/list/deque).
Уметь	Читать/записывать структурированные данные в файлы.
	Реализовывать алгоритмы на графах (матрица смежности vs список ребер).
	Использовать итераторы и лямбды для обработки данных.
Владеть	Обработкой больших данных с помощью файлового ввода-вывода.
	Применением алгоритмов STL (sort, find, accumulate).
	Реализацией графовых алгоритмов на C++.
ОПК-2	Способен применять современный математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности
ИД-1.ОПК-2	Способен применять системный подход к анализу предметной (проблемной) области, выявлению требований к ИС
Знать	Математические основы алгоритмов (теория графов, асимптотический анализ).

Принципы работы с памятью в C++ (указатели, динамические структуры).

- Уметь** Анализировать требования и выбирать подходящую структуру.
Использовать шаблоны проектирования (итератор, адаптер).
- Владеть** Реализацией структур данных "с нуля" (например, собственный *LinkedList*).
Сравнением *STL* и рукописных решений.

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

- Знать** Расширенные возможности C++ (шаблоны, *move*-семантика, многопоточность).
Алгоритмы сортировки и поиска для деревьев и графов.
- Уметь** Оптимизировать код (кеширование, эффективные обходы).
Тестировать и отлаживать сложные структуры.
- Владеть** Реализацией деревьев (бинарные, *AVL*) и графовых алгоритмов.
Использованием *STL* для высокопроизводительных вычислений.

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

УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач

ИД-1.УК-1 Осуществляет поиск необходимой информации, опираясь на результаты анализа поставленной задачи

- Знать** Современные методы программирования на C++ (*ООП*, *STL*, шаблоны) на уровне, достаточном для решения стандартных задач.
Принципы работы и реализации:
Линейных динамических структур (односвязные, двусвязные, кольцевые списки) с пониманием их основных операций и сложности.
Деревьев (бинарные, *AVL*, красно-черные) на уровне понимания базовых алгоритмов обхода и балансировки.
Графов (алгоритмы обхода, поиска кратчайшего пути) с возможностью их реализации в стандартных случаях.
Контейнеров *STL* (*vector*, *list*, *map*, *set* и др.) и их основных характеристик.
- Уметь** Анализировать требования к структурам данных и выбирать оптимальную реализацию из стандартных вариантов.
Проверять корректность реализации алгоритмов (верификация кода) с использованием базовых методов тестирования.
Применять методы анализа эффективности алгоритмов (*O*-нотация) для сравнения стандартных решений.
- Владеть** Разработкой и оптимизацией структур данных на C++ в рамках типовых задач курса.
Сравнительным анализом различных реализаций (например, *std::list* vs рукописный список) с обоснованием выбора.
Обработкой данных из файлов и их структурированным хранением с использованием стандартных подходов.
Применением *STL*-контейнеров и алгоритмов для решения задач в соответствии с изученными методиками.

ИД-2.УК-1 Выбирает оптимальный вариант решения задачи, аргументируя свой выбор

Знать	Преимущества и недостатки различных структур данных (массивы vs списки, хеш-таблицы vs деревья) на уровне, позволяющем делать осознанный выбор для типовых задач. Особенности работы с памятью в C++ (указатели, аллокация, кэш-локальность) и их влияние на производительность. Критерии выбора алгоритмов (временная и пространственная сложность) для стандартных случаев
Уметь	Сравнивать различные подходы к реализации (например, итеративный vs рекурсивный обход дерева) и выбирать оптимальный для заданных условий. Обосновывать выбор контейнера STL для конкретной задачи с учетом требований к производительности. Анализировать производительность кода с помощью профилирования и вносить базовые оптимизации.
Владеть	Оптимизацией структур данных под конкретные задачи (кеширование, пулы памяти) в рамках изученных методов. Реализацией высокоэффективных алгоритмов на графах и деревьях для стандартных сценариев. Использованием продвинутых возможностей STL (аллокаторы, пользовательские компараторы) в соответствии с изученными примерами.
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
ИД-3.УК-2	Использует принципы проектной методологии для решения профессиональных задач
Знать	Алгоритмы и структуры данных в расширенном объеме Критерии выбора решений для различных случаев
Уметь	Выбирать между структурами данных с учетом их характеристик Анализировать производительность различными методами
Владеть	Реализацией динамических структур с контролем ресурсов Применением STL в сложных сценариях
ИД-4.УК-2	Выбирает оптимальный способ решения задач, имеющихся ресурсов и ограничений, оценки рисков на основе проектного инструментария
Знать	Методы работы с файлами в различных режимах Алгоритмы обработки графов и их модификации
Уметь	Реализовывать алгоритмы с вариативными параметрами Использовать STL для сложных операций
Владеть	Обработкой данных в различных форматах Реализацией алгоритмов с контролем точности
ОПК-2	Способен применять современный математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности
ИД-1.ОПК-2	Способен применять системный подход к анализу предметной (проблемной) области, выявлению требований к ИС
Знать	Математические основы и их программную реализацию Принципы работы с памятью на продвинутом уровне
Уметь	Анализировать требования и предлагать оптимальные решения

Владеть Использовать шаблоны проектирования в сложных случаях
 Реализацией структур данных с высокой эффективностью
 Сравнением подходов с профессиональной точки зрения

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

Знать Расширенные возможности C++ и их практическое применение
 Алгоритмы обработки данных в профессиональном контексте

Уметь Оптимизировать код с учетом различных критериев
 Тестировать реализации различными методами

Владеть Реализацией сложных структур и алгоритмов
 Использованием STL на профессиональном уровне

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

УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач

ИД-1.УК-1 Осуществляет поиск необходимой информации, опираясь на результаты анализа поставленной задачи

Знать Современные методы программирования на C++ на экспертном уровне, включая тонкости реализации и скрытые возможности
 Принципы работы структур данных с глубоким пониманием их внутреннего устройства и возможностей кастомизации
 Методы работы с файлами и их профессиональное использование в сложных системах

Уметь Анализировать требования и разрабатывать высокоэффективные и масштабируемые решения
 Проверять корректность реализации с применением продвинутых методов тестирования и верификации
 Применять анализ эффективности для глубокой оптимизации кода и выявления узких мест

Владеть Разработкой структур данных с максимальной эффективностью и учетом всех аспектов производительности
 Сравнительным анализом реализаций с обоснованием выбора на основе комплексного тестирования
 Профессиональной работой с файлами включая обработку больших объемов данных и сложные форматы

ИД-2.УК-1 Выбирает оптимальный вариант решения задачи, аргументируя свой выбор

Знать Характеристики структур данных на уровне эксперта, включая нюансы реализации в различных условиях
 Особенности работы с памятью и продвинутые методы оптимизации памяти

Уметь Сравнивать подходы к реализации и разрабатывать гибридные решения, сочетающие преимущества разных методов
 Обосновывать выбор инструментов с учетом всех возможных сценариев использования и потенциальных рисков

Владеть Глубокой оптимизацией структур данных для экстремальных условий работы

	<i>Реализацией алгоритмов с гарантированной эффективностью и минимальными накладными расходами</i>
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
ИД-3.УК-2	Использует принципы проектной методологии для решения профессиональных задач
Знать	Алгоритмы и структуры данных на уровне, позволяющем модифицировать и улучшать стандартные реализации Критерии выбора решений для экстремальных и нестандартных случаев
Уметь	Выбирать между структурами данных и создавать комбинированные структуры для специфических задач Анализировать производительность с использованием профессиональных инструментов и методик
Владеть	Реализацией динамических структур с экстремальной эффективностью и надежностью Профессиональным применением STL включая создание специализированных аллокаторов и адаптеров
ИД-4.УК-2	Выбирает оптимальный способ решения задач, имеющихся ресурсов и ограничений, оценки рисков на основе проектного инструментария
Знать	Методы работы с файлами на профессиональном уровне, включая работу с потоками и асинхронными операциями Алгоритмы обработки графов и их специализированные модификации
Уметь	Реализовывать алгоритмы с дополнительными оптимизациями для конкретных аппаратных архитектур Использовать STL для создания высокопроизводительных специализированных решений
Владеть	Профессиональной обработкой данных в реальном времени и в распределенных системах Реализацией алгоритмов с гарантированной точностью и отказоустойчивостью
ОПК-2	Способен применять современный математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности
ИД-1.ОПК-2	Способен применять системный подход к анализу предметной (проблемной) области, выявлению требований к ИС
Знать	Математические основы на уровне, позволяющем разрабатывать новые алгоритмы Принципы работы с памятью и методы создания специализированных систем управления памятью
Уметь	Анализировать требования и предлагать инновационные архитектурные решения Использовать шаблоны проектирования для создания расширяемых и поддерживаемых систем

Владеть	Созданием структур данных с уникальными характеристиками под конкретные задачи Профессиональным сравнением подходов с разработкой новых критериев оценки
ИД-3.ОПК-2	Аргументировано применяет методы проектирования, разработки и реализации программных продуктов и программных комплексов в различных областях человеческой деятельности
Знать	Расширенные возможности C++ на уровне эксперта, включая метапрограммирование и низкоуровневую оптимизацию Алгоритмы обработки данных и методы их адаптации для специализированных применений
Уметь	Оптимизировать код на уровне процессорных архитектур и кэш-памяти Тестировать реализации с использованием профессиональных методологий и инструментов
Владеть	Созданием сложных структур данных и алгоритмов производственного уровня Профессиональным использованием STL с расширением его функциональности

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

Типовые контрольные задания:

1. Описать функцию, которая вставляет в список L за первым вхождением элемента E все элементы списка L1, если E входит в L.
2. Описать функцию, которая формирует список L, включив в него по одному разу элементы, которые входят в один из списков L1 и L2, но в то же время не входят в другой из них;
3. Описать функцию, которая подсчитывает количество чисел списка L, у которых первая и последняя цифра совпадают.
4. Описать функцию, которая вычисляет значение многочлена в некоторой целочисленной точке.
5. Дана последовательность целых чисел, оканчивающаяся нулем. Сформировать двунаправленный список на основе данной последовательности. Найти количество элементов, значения которых больше заданного.
6. Дана последовательность целых чисел, оканчивающаяся нулем. Сформировать двунаправленный список на основе данной последовательности. Удалить из списка все максимальные элементы.
7. Дана последовательность целых чисел, оканчивающаяся нулем. Сформировать двунаправленный список на основе данной последовательности. Расположить отрицательные элементы вначале списка с сохранением их исходного порядка ввода.
8. Дана последовательность целых чисел, оканчивающаяся нулем. Сформировать однонаправленный кольцевой список на основе данной последовательности. Найти в списке наиболее длинную убывающую подпоследовательность.
9. Дана последовательность целых чисел, оканчивающаяся нулем. Сформировать однонаправленный кольцевой список на основе данной

последовательности. Найти в списке наиболее длинную подпоследовательность, элементы которой меньше среднего арифметического значений всех элементов списка.

10. Дана последовательность целых чисел, оканчивающаяся нулем. Написать программную реализацию хранения данной последовательности в виде дерева двоичного поиска. Написать функцию, подсчитывающую сумму элементов дерева.

11. Дана последовательность целых чисел, оканчивающаяся нулем. Написать программную реализацию хранения данной последовательности в виде дерева двоичного поиска. Написать функцию, которая находит наибольший элемент дерева.

12. Дана последовательность целых чисел, оканчивающаяся нулем. Написать программную реализацию хранения данной последовательности в виде дерева двоичного поиска. Найти количество листьев дерева.

13. Каждая запись текстового файла содержит текст на английском языке. Требуется написать программу, которая для каждой записи файла будет определять, можно ли переставить латинские буквы каждой записи так, чтобы получился палиндром (палиндром читается одинаково слева направо и справа налево). В выходном файле записать ответ «Да» или «Нет», а в случае ответа «Да» – еще и сам полученный палиндром (первый в алфавитном порядке).

14. Каждая запись входного текстового файла содержит последовательность цифр. Написать программу, которая цифры, встречающиеся во входной записи, в порядке увеличения частоты их встречаемости выводит в соответствующую запись выходного файла. Если какие-то цифры встречаются одинаковое число раз, они выводятся в порядке возрастания.

15. Дан целочисленный вектор размера N . Вычислить минимальное значение элементов вектора; умножить каждый элемент вектора на 5 и добавить в конец вектора элемент, равный вычисленному минимуму.

16. Сформировать множество из 20 случайных чисел. Определить, входят ли в это множество 2 заданных пользователем числа.

17. Дана строка. Если длина строки нечетна, то удалить средний символ, иначе вставить в середину строки «abc».

18. Дана строка. Если в строке больше двух цифр, то удалить первый и последний символы строки,

19. Дан связанный неориентированный граф. Найти цикл максимальной длины.

20. Дан связанный неориентированный нагруженный граф. Найти минимальное остовное дерево.

Типовые задачи для самостоятельного решения:

1. Описать функцию, которая переворачивает список L , т.е. изменяет ссылки в этом списке так, чтобы его элементы оказались расположенными в обратном порядке;

2. Описать функцию, которая подсчитывает число вхождений элемента E в список L .

3. Описать функцию, которая формирует список L , включив в него по одному разу элементы, которые входят хотя бы в один из списков $L1$ и $L2$;

4. Описать функцию, которая удаляет из непустых слов списка L их первые буквы.

5. Описать функцию, которая формирует список L , включив в него по одному разу элементы, которые входят одновременно в оба списка $L1$ и $L2$;

6. Описать функцию, которая определяет количество слов в непустом списке L , отличных от последнего.

7. Описать функцию, которая объединяет два упорядоченных по неубыванию списка $L1$ и $L2$ в один упорядоченный по неубыванию список, меняя соответствующим образом ссылки в $L1$ и $L2$ и присвоив полученный список параметру $L1$.

8. Описать функцию, которая в списке L заменяет первое вхождение списка L1 (если такое есть) на список L2.
9. Описать функцию, которая проверяет на равенство два многочлена (указывается коэффициент и степень каждого элемента).
10. Описать функцию, которая строит многочлен p — сумму многочленов q и r .
11. Дана последовательность целых чисел, оканчивающаяся нулем. Сформировать двунаправленный список на основе данной последовательности. Найти значение минимального элемента в списке.
12. Дана последовательность целых чисел, оканчивающаяся нулем. Сформировать двунаправленный список на основе данной последовательности. Найти значение среднего арифметического нечетных отрицательных элементов.
13. Дана последовательность целых чисел, оканчивающаяся нулем. Сформировать двунаправленный список на основе данной последовательности. Найти количество повторяющихся элементов.
14. Дана последовательность целых чисел, оканчивающаяся нулем. Сформировать двунаправленный список на основе данной последовательности. Найти количество неповторяющихся четных элементов.
15. Дана последовательность целых чисел, оканчивающаяся нулем. Сформировать двунаправленный список на основе данной последовательности. Упорядочить элементы списка по убыванию.
16. Дана последовательность целых чисел, оканчивающаяся нулем. Сформировать двунаправленный список на основе данной последовательности. Проходя список слева направо, после каждого второго элемента добавить элемент, значение которого есть разность двух предыдущих элементов.
17. Дана последовательность целых чисел, оканчивающаяся нулем. Сформировать двунаправленный список на основе данной последовательности. Удалить из списка все элементы с нечетными значениями.
18. Дана последовательность целых чисел, оканчивающаяся нулем. Сформировать двунаправленный список на основе данной последовательности. Определить, есть ли в списке два равных элемента, расположенных на диаметре кольца.
19. Дана последовательность целых чисел, оканчивающаяся нулем. Сформировать однонаправленный кольцевой список на основе данной последовательности. Определить, можно ли разделить список на две части с равным количеством элементов таких, что суммы их элементов равны.
20. Дана последовательность целых чисел, оканчивающаяся нулем. Сформировать однонаправленный кольцевой список на основе данной последовательности. Найти в списке наиболее длинную возрастающую подпоследовательность.
21. Дана последовательность целых чисел, оканчивающаяся нулем. Сформировать двунаправленный кольцевой список на основе данной последовательности. Найти в списке наиболее длинную симметричную подпоследовательность.
22. Дана последовательность целых чисел, оканчивающаяся нулем. Сформировать двунаправленный кольцевой список на основе данной последовательности. Найти в списке наиболее длинную подпоследовательность, состоящую из элементов с нечетной суммой цифр.
23. Дана последовательность целых чисел, оканчивающаяся нулем. Написать программную реализацию хранения данной последовательности в виде дерева двоичного поиска. Написать функцию, которая находит наименьший четный элемент дерева.
24. Дана последовательность целых чисел, оканчивающаяся нулем. Написать программную реализацию хранения данной последовательности в виде дерева двоичного поиска. Заменить четные элементы дерева нулем.

25. Дана последовательность целых чисел, оканчивающаяся нулем. Написать программную реализацию хранения данной последовательности в виде дерева двоичного поиска. Написать функцию, которая выводит на экран элементы из всех листьев дерева.

26. Дана последовательность целых чисел, оканчивающаяся нулем. Написать программную реализацию хранения данной последовательности в виде дерева двоичного поиска. Написать функцию, которая определяет глубину (уровень) заданного элемента дерева, информационное поле которого равно заданному значению, и возвращает -1 , если такого элемента нет.

27. Дана последовательность целых чисел, оканчивающаяся нулем. Написать программную реализацию хранения данной последовательности в виде дерева двоичного поиска. Написать функцию, которая находит глубину дерева.

28. Дана последовательность целых чисел, оканчивающаяся нулем. Написать программную реализацию хранения данной последовательности в виде дерева двоичного поиска. Написать функцию, которая по заданному числу n определяет количество всех вершин, находящихся на уровне с номером n в дереве.

29. Дан одномерный массив. Отсортировать массив, используя дерево двоичного поиска.

30. Дана последовательность символов. Определить частоту вхождения каждого из символов в последовательность, используя дерево двоичного поиска.

31. Дана последовательность целых чисел, оканчивающаяся нулем. Написать программную реализацию хранения данной последовательности в виде дерева двоичного поиска. Вывести количество вершин дерева, значение которых не кратно заданному числу K .

32. Дана последовательность целых чисел, оканчивающаяся нулем. Написать программную реализацию хранения данной последовательности в виде дерева двоичного поиска. Найти сумму нечетных значений всех листьев дерева.

33. Дана последовательность целых чисел, оканчивающаяся нулем. Написать программную реализацию хранения данной последовательности в виде дерева двоичного поиска.

34. Радиотелескоп пытается получать и анализировать сигналы из космоса. Все сигналы представляются в виде вещественного неотрицательного числа и сохраняются последовательно в текстовом файле. Характеристикой определенного района космоса считается число, равное максимальному произведению, которое можно получить, перемножая значения сигналов, приходящих из этого района. То есть требуется выбрать такое непустое подмножество сигналов (в него может войти как один сигнал, так и все поступившие сигналы), произведение значений у которого будет максимальным. Если таких подмножеств несколько, то выбрать можно любое из них. Напишите программу, которая будет обрабатывать результаты эксперимента, находя искомое подмножество.

35. Файл содержит текст заклинания. Гарри Поттеру нужно зашифровать его следующим образом. В каждом абзаце Гарри сначала определяет количество букв в самом коротком слове, обозначив полученное число через K . Затем он заменяет каждую английскую букву в заклинании на букву, стоящую в английском алфавите на K букв ранее (алфавит считается циклическим, т. е. перед буквой A стоит буква Z), оставив другие символы неизменными. Строчные буквы при этом остаются строчными, а прописные – прописными. Написать программу для Гарри Поттера, которая будет выводить в файл текст зашифрованного заклинания.

36. Файл содержит результаты голосования избирателей за несколько партий, в виде списка названий данных партий. В каждой строке файла записано название партии, за которую проголосовал данный избиратель. Название партии может содержать буквы, цифры, пробелы и прочие символы. Количество партий определяется на основе данных файла. Программа должна вывести в выходной файл список названий всех партий, встречающихся в исходном списке, в порядке убывания количества голосов, отданных за

эту партию. При этом название каждой партии должно быть выведено ровно один раз, вне зависимости от того, сколько голосов было отдано за данную партию.

37. Входной файл содержит текст, в котором нужно зашифровать все английские слова. Каждое слово шифруется с помощью циклического сдвига на длину этого слова. Например, если длина слова равна K , каждая буква в слове заменяется на букву, стоящую в английском алфавите на K букв дальше (алфавит считается циклическим, т. е. за буквой Z стоит буква A). Строчные буквы при этом остаются строчными, а прописные – прописными. Символы, не являющиеся английскими буквами, не изменяются. Зашифрованный текст разместить в выходном файле.

38. Каждая запись входного файла содержит последовательность символов. Написать программу, которая определяет, есть ли в этой последовательности десятичные цифры, и формирует наибольшее число, которое можно составить из этих цифр. Ведущих нулей в числе быть не должно (за исключением числа 0, запись которого содержит ровно одну цифру). Если цифр нет, программа должна вывести в выходной файл сообщение «NO», а если есть – полученное число.

39. Дан целочисленный вектор размера N . Увеличить все четные числа, содержащиеся в векторе, на исходное значение первого четного числа. Если четные числа в векторе отсутствуют, то оставить вектор без изменений.

40. Дан вектор A размера N и целые числа K и L ($1 \leq K < L \leq N$). Переставить в обратном порядке элементы вектора, расположенные между элементами $A[K]$ и $A[L]$, не включая эти элементы.

41. Дан целочисленный вектор размера N . Удалить из вектора все элементы, встречающиеся более одного раза, и вывести размер полученного вектора и его содержимое.

42. Создать множество чисел (`set <int>`), записать в него 20 различных чисел, а затем удалить из множества все числа больше 10.

43. Создать словарь (`map <string, string>`), занести в него десять записей по принципу «Фамилия» – «Имя», после чего проверить, сколько людей имеют совпадающие с заданным имя или фамилию.

44. Создать словарь (`map <string, string>`), занести в него десять записей по принципу «Фамилия» – «Имя». Удалить из словаря людей с повторяющимися именами.

45. Дан набор целых чисел с четным количеством элементов. Заполнить вектор V исходными числами и вывести вначале вторую половину элементов вектора V , а затем первую половину (в каждой половине порядок элементов не изменять).

46. Дан набор целых чисел с четным количеством элементов. Заполнить дек D исходными числами и вывести первую половину элементов дека D в обратном порядке, а затем — вторую половину (также в обратном порядке).

47. Дан набор целых чисел, количество которых делится на 3. Заполнить список L исходными числами и вывести вначале первую треть элементов списка L в исходном порядке, затем вторую треть элементов в обратном порядке, а затем последнюю треть (также в обратном порядке).

48. Дана строка. Из второй половины каждой строки удалить все «.».

49. Дана строка. Если длина строки четна, то вставить в середину строки первый символ.

50. Дана строка. Заменить в строке все цифры на первую букву строки.

51. Дана строка S . Исключить из строки группы символов, расположенные между скобками (,). Сами скобки тоже должны быть исключены. Предполагается, что внутри каждой пары скобок нет других скобок.

52. Дана строка. Если буква «а» входит в первую половину строки, то вставить в начало строки «abc», иначе удалить первый символ строки.

53. Дан связанный неориентированный нагруженный граф. Найти две наиболее удаленные вершины.

54. Дан связанный неориентированный граф. Определить, есть ли в нем цикл.
55. Дан связанный неориентированный нагруженный граф. Найти кратчайшее расстояние между всеми парами вершин. Дан связанный ориентированный граф. Определить, описывает ли данный граф отношение частичного порядка.
56. Дан связанный ориентированный граф. Определить, описывает ли данный граф отношение линейного порядка.
57. Дан неориентированный граф. Найти компоненту связности с наибольшим количеством вершин.

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

Вопросы для подготовки к экзамену

1. Способы организации распределения памяти
2. Динамические структуры данных
3. Классификация динамических структур данных
4. Однонаправленные (односвязные) списки
5. Вставка элемента в однонаправленный список
6. Удаление элемента из однонаправленного списка
7. Двухнаправленные (двусвязные) списки
8. Удаление элемента из двухнаправленного списка
9. Вставка элемента в двухнаправленный список
10. Рекурсия в связных списках
11. Кольцевые списки
12. Стеки
13. Реализация стека с помощью одномерного массива
14. Реализация стека с помощью динамических структур данных
15. Очереди
16. Реализация очереди с помощью одномерного массива
17. Реализация очереди с помощью динамических структур данных
18. Деревья. Основные понятия
19. Бинарные деревья. Классификация бинарных деревьев
20. Обходы деревьев
21. Двоичное дерево поиска. Построение.
22. Потоки ввода и вывода. Стандартные потоки в C++.
23. Базовый файловый ввод и вывод. Буферизованный вывод
24. Режимы открытия файлов. Файловый указатель. Прямой доступ к файлам
25. Стандартная библиотека шаблонов (STL)
26. Класс string. Основные функции
27. Контейнеры. Основные понятия. Виды контейнеров.
28. Итераторы. Объявление и использование итераторов. Прямые и обратные итераторы.
29. Контейнер vector. Основные функции работы с vector
30. Контейнер deque. Основные функции работы с deque
31. Контейнер list. Основные функции работы с list
32. Контейнер forward_list. Основные функции работы с forward_list
33. Контейнер stack. Основные функции работы с stack
34. Контейнер set. Основные функции работы с set
35. Контейнер map. Основные функции работы с map
36. Графы. Основные понятия. Способы представления графа
37. Графы. Поиск в глубину
38. Графы. Поиск в ширину

39. Алгоритм Дейкстры
40. Алгоритм Флойда
41. Построение минимального остовного дерева. Алгоритм Краскала
42. Построение минимального остовного дерева. Алгоритм Прима

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

Методические рекомендации, определяющие процедуры оценивания выполнения контрольных заданий:

Задание считается выполненным при выполнении следующих условий:

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

Методические рекомендации, определяющие процедуры оценивания на зачете:

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

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

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

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

Критерии оценки:

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

Методические рекомендации к сдаче экзамена

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

Текущий контроль успеваемости студентов проводится в целях совершенствования и непрерывного контроля качества образовательного процесса, проверки усвоения учебного

материала, активизации самостоятельной работы студентов, стимулирования их учебной работы, обеспечения эффективности образовательного процесса, предупреждения рисков отчисления студентов.

Текущий контроль знаний студентов осуществляется постоянно в течение всего семестра.

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

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

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

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

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

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

На зачете/экзамене могут присутствовать ректор, проректор по учебной работе, декан факультета, заведующий кафедрой, которая обеспечивает учебный процесс по данной дисциплине. Присутствие на экзаменах и зачетах посторонних лиц без разрешения ректора или проректора по учебной работе не допускается.

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

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

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

Методические рекомендации, определяющие процедуры оценивания на экзамене:

Критерии оценивания и шкала оценки экзамена

Оценка	Критерии выставления оценки
«Отлично»	Дан развернутый ответ на поставленные вопросы. Материал изложен последовательно. Имеются логичные и аргументированные выводы. В течении семестра студент правильно решил 80 % задач, имеет представление как решать остальные задачи
«Хорошо»	Дан развернутый ответ на поставленные вопросы. Материал изложен в целом последовательно. Имеются логичные и аргументированные выводы.

	В течении семестра студент правильно решил 70 % задач, имеет представление как решать остальные задачи
«Удовлетворительно»	Ответ на вопрос не является полным. Материал изложен непоследовательно. Выводы не аргументированы. В течении семестра студент правильно решил 60 % задач, имеет представление как решать остальные задачи
«Неудовлетворительно»	Обучающийся не знает значительной части программного материала, допускает существенные ошибки. В течении семестра студент правильно решил менее 70 % задач и/или не имеет представление как решать остальные задачи

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

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

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

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

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

Для лиц с нарушениями зрения:

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

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

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

5.1 Основная литература:

1. Конова, Е. А. Алгоритмы и программы. Язык C++ : учебное пособие для вузов / Е. А. Конова, Г. А. Поллак. — 6-е изд., стер. — Санкт-Петербург : Лань, 2021. — 384 с. — ISBN 978-5-8114-8487-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/176900> (дата обращения: 08.06.2025). — Режим доступа: для авториз. пользователей.

2. Огнева, М. В. Программирование на языке C++: практический курс : учебное пособие для вузов / М. В. Огнева, Е. В. Кудрина. — Москва : Издательство Юрайт, 2022. — 335 с. — (Высшее образование). — ISBN 978-5-534-05123-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/492984> (дата обращения: 08.06.2025).

3. Кувшинов, Д. Р. Основы программирования : учебное пособие для вузов / Д. Р. Кувшинов. — Москва : Издательство Юрайт, 2022. — 104 с. — (Высшее образование). — ISBN 978-5-534-07559-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/493460> (дата обращения: 08.06.2025).

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

5.2 Дополнительная литература:

1. Страуструп, Б. Язык программирования C++ для профессионалов : учебное пособие / Б. Страуструп. — 2-е изд. — Москва : ИНТУИТ, 2016. — 670 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100542> (дата обращения: 08.06.2025). — Режим доступа: для авториз. пользователей.

2. Павловская, Т. А. Программирование на языке C++ : учебное пособие / Т. А. Павловская. — 2-е изд. — Москва : ИНТУИТ, 2016. — 154 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100409> (дата обращения: 08.06.2025). — Режим доступа: для авториз. пользователей.

3. Керниган, Б. В. Язык программирования C : учебник / Б. В. Керниган, Д. М. Ричи. — 2-е изд. — Москва : ИНТУИТ, 2016. — 313 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100543> (дата обращения: 08.06.2025). — Режим доступа: для авториз. пользователей.

4.

5.3. Периодические издания:

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

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

Электронно-библиотечные системы (ЭБС):

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

Профессиональные базы данных

1. Scopus <http://www.scopus.com/>
2. ScienceDirect <https://www.sciencedirect.com/>
3. Журналы издательства Wiley <https://onlinelibrary.wiley.com/>
4. Научная электронная библиотека (НЭБ) <http://www.elibrary.ru/>
5. Полнотекстовые архивы ведущих западных научных журналов на Российской платформе научных журналов НЭИКОН <http://archive.neicon.ru>
6. Национальная электронная библиотека (доступ к Электронной библиотеке диссертаций Российской государственной библиотеки (РГБ) <https://rusneb.ru/>
7. Президентская библиотека им. Б.Н. Ельцина <https://www.prlib.ru/>
8. База данных CSD Кембриджского центра кристаллографических данных (CCDC) <https://www.ccdc.cam.ac.uk/structures/>
9. Springer Journals: <https://link.springer.com/>
10. Springer Journals Archive: <https://link.springer.com/>
11. Nature Journals: <https://www.nature.com/>

12. Springer Nature Protocols and Methods: <https://experiments.springernature.com/sources/springer-protocols>
13. Springer Materials: <http://materials.springer.com/>
14. Nano Database: <https://nano.nature.com/>
15. Springer eBooks (i.e. 2020 eBook collections): <https://link.springer.com/>
16. "Лекториум ТВ" <http://www.lektorium.tv/>
17. Университетская информационная система РОССИЯ <http://uisrussia.msu.ru>

Информационные справочные системы

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

Ресурсы свободного доступа

1. КиберЛенинка <http://cyberleninka.ru/>;
2. Американская патентная база данных <http://www.uspto.gov/patft/>
3. Министерство науки и высшего образования Российской Федерации <https://www.minobrnauki.gov.ru/>;
4. Федеральный портал "Российское образование" <http://www.edu.ru/>;
5. Информационная система "Единое окно доступа к образовательным ресурсам" <http://window.edu.ru/>;
6. Единая коллекция цифровых образовательных ресурсов <http://school-collection.edu.ru/> .
7. Проект Государственного института русского языка имени А.С. Пушкина "Образование на русском" <https://pushkininstitute.ru/>;
8. Справочно-информационный портал "Русский язык" <http://gramota.ru/>;
9. Служба тематических толковых словарей <http://www.glossary.ru/>;
10. Словари и энциклопедии <http://dic.academic.ru/>;
11. Образовательный портал "Учеба" <http://www.ucheba.com/>;
12. Законопроект "Об образовании в Российской Федерации". Вопросы и ответы http://xn--273--84d1f.xn--p1ai/voprosy_i_otvety

Собственные электронные образовательные и информационные ресурсы КубГУ

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

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

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

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

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

7.1 Перечень информационно-коммуникационных технологий

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

7.2 Перечень лицензионного и свободно распространяемого программного обеспечения

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

8. Материально-техническое обеспечение по дисциплине (модулю)

№	Вид работ	Наименование учебной аудитории, ее оснащенность оборудованием и техническими средствами обучения
1.	Лекционные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения
2.	Лабораторные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, проектором, программным обеспечением
3.	Групповые (индивидуальные) консультации	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
4.	Текущий контроль, промежуточная аттестация	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
5.	Самостоятельная работа	Кабинет для самостоятельной работы, оснащенный компьютерной техникой с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета.

Примечание: Конкретизация аудиторий и их оснащение определяется ОПОП.