

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Московский физико-технический институт  
(национальный исследовательский университет)»**

**УТВЕРЖДЕНО**

**Директор физтех-школы  
биологической и медицинской  
физики**

**Д.В. Кузьмин**

**Рабочая программа дисциплины (модуля)**

<b>по дисциплине:</b>	Алгоритмы и структуры данных
<b>по направлению:</b>	Прикладные математика и физика
<b>профиль подготовки:</b>	Алгоритмическая биология Физтех-школа Биологической и Медицинской Физики центр образовательных программ Физтех-школы биологической и медицинской физики
<b>курс:</b>	1
<b>квалификация:</b>	магистр

Семестр, формы промежуточной аттестации: 1 (осенний) - Экзамен

Аудиторных часов: 90 всего, в том числе:

лекции: 30 час.

семинары: 60 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 60 час.

Подготовка к экзамену: 30 час.

Всего часов: 180, всего зач. ед.: 4

Программу составил: А.С. Дух, старший преподаватель

Программа обсуждена на заседании центра образовательных программ Физтех-школы биологической и медицинской физики 16.06.2023

## Аннотация

Дисциплина направлена на освоение алгоритмов и структур данных на языке C++. После изучения курса студенты научатся составлять алгоритмы обхода графов, понимать работу библиотек и активно использовать их. Домашние задания по курсу закрепляют полученные знания и воспитывают хороший стиль написания кода, который позволяет избежать стандартных ошибок.

### 1. Цели и задачи

#### Цель дисциплины

- дать студентам базовые знания в области алгоритмов и структур данных, важные для понимания работы библиотек, алгоритмов и языков программирования.

#### Задачи дисциплины

1. Познакомиться с основными алгоритмами и структурами данных поиска.
2. Получить представление о проблемах, возникающих при применении известных алгоритмов анализа данных для решения практических задач поиска. Научиться преодолевать эти сложности имеющимися в распоряжении средствами.
3. Научиться оценивать учетную стоимость операций и алгоритмическую сложность кода.
4. Изучить задачи сортировки, модели вычислений, структуры данных с хранением истории, деревья поиска, задачи о динамическом поиске, алгоритмы обхода графов, поиска кратчайших путей, задачи подстроки в строке.
5. Получить практический опыт программирования, выработать хороший стиль написания кода, который позволяет избежать стандартных, но от этого ничуть не менее распространенных даже у опытных разработчиков ошибок.

### 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-2 Имеет представление об актуальных проблемах науки и техники в области своей профессиональной деятельности, способен на научном языке формулировать профессиональные задачи	ОПК-2.1 Имеет представление о современном состоянии исследований в рамках тематической области своей профессиональной деятельности
	ОПК-2.2 Способен оценивать актуальность исследований в области своей профессиональной деятельности и их практическую значимость
ОПК-4 Способен успешно реализовывать решение поставленной задачи, провести анализ результата и представить выводы, применяя знания и навыки в области физико-математических наук и информационно-коммуникационных технологий	ОПК-4.1 Способен применять знания и навыки по использованию информационно-коммуникационных технологий для поиска и изучения научной литературы, применения прикладных программных продуктов
	ОПК-4.2 Способен применять знания в области физико-математических наук для решения поставленной задачи, формулирования выводов и оценки полученных результатов

### 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

Основные алгоритмы анализа данных, их преимущества и недостатки, а также структуры данных поиска.

уметь:

Использовать средства языка программирования C++ для разработки надежных и быстро работающих программных систем. Создавать качественный код для реализации алгоритмов анализа данных.

владеть:

Средствами разработки и тестирования программного кода на языке C++.

#### 4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

##### 4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Понятие базы данных	2	4		3
2	Аминокислотная последовательность белка	2	4		3
3	Знакомство с базами данных нуклеотидных последовательностей, их история	2	4		3
4	BLAST: краткое теоретическое введение	2	4		3
5	Биологические структуры, база Protein Data Bank	2	4		3
6	Краткие принципы биологической классификации и таксономии	2	4		3
7	Понятие метаболических путей, сигнальных путей	2	4		3
8	База данных OMIM	2	4		3
9	Сложность и модели вычислений. Анализ учетных стоимостей.	2	4		5
10	Алгоритмы Merge-Sort и Quick-Sort.	2	4		5
11	Порядковые статистики. Кучи. Хэширование	2	4		5
12	Деревья поиска. Система непересекающихся множеств. Задачи RMQ и LCA. Структуры данных для геометрического поиска.	2	4		5
13	Поиск кратчайших путей. Минимальные остовные деревья. Минимальные разрезы.	2	4		5
14	Поиск подстрок. Длиннейшие общие подстроки. Приближенный поиск подстрок.	2	4		5
15	Суффиксные деревья. Суффиксные массивы.	2	4		6
Итого часов		30	60		60
Подготовка к экзамену		30 час.			
Общая трудоёмкость		180 час., 4 зач.ед.			

##### 4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 1 (Осенний)

###### 1. Понятие базы данных

Какие биологические данные можно хранить в базах данных? Обзор основных существующих биологических баз данных. Обзор форматов биологических данных. Как конвертировать разные форматы биологических данных?

## 2. Аминокислотная последовательность белка

Знакомство с базой данных Uniprot, продвинутый поиск в Uniprot. База данных Pfam. Практическое задание: выжать как можно сведений о некотором белке с помощью упомянутых баз данных. Работа с выравниваниями, загруженными из Pfam, с помощью программы JalView. Понятие филогенетического дерева. Извлечение филогенетических деревьев из базы Pfam. Работа с филогенетическими деревьями в программе MEGA7.

## 3. Знакомство с базами данных нуклеотидных последовательностей, их история

Знакомство с различными базами данных NCBI. Работа с выравниваниями нуклеотидных последовательностей в программе JalView. Работа с геномным браузером.

## 4. BLAST: краткое теоретическое введение

Практические задания по работе с BLAST: разные виды BLAST, управление параметрами BLAST, переключение на разные базы данных.

## 5. Биологические структуры, база Protein Data Bank

Структура файла формата .pdb, основы работы в PyMol, построение пространственного выравнивания с помощью PDBeFold.

## 6. Краткие принципы биологической классификации и таксономии

Краткие принципы биологической классификации и таксономии. Знакомство с некоторыми таксономическими базами данных (IT IS, NCBI, EOL).

## 7. Понятие метаболических путей, сигнальных путей

Понятие метаболических путей, сигнальных путей. Функциональные базы данных: GO, KEGG.

## 8. База данных OMIM

Базы мутаций. База данных PubMed. Как искать биологическую литературу с помощью PubMed и Google Scholar?

## 9. Сложность и модели вычислений. Анализ учетных стоимостей.

Основные ресурсы: память и время. O-символика. Примеры моделей вычисления: машина Тьюринга, RAM-машина. Сложность в среднем и худшем случаях. Пример: задача сортировки. Сортировка выбором. Теоретико-информационная нижняя оценка сложности. Разрешающие деревья. Нижняя оценка сложности в модели разрешающих деревьев. Массивы переменного размера: аддитивная и мультипликативная схемы реаллокации. Анализ мультипликативной схемы для массива переменного размера с помощью банковского метода. Анализ учетных стоимостей операций: функция потенциала, истинные и учетные стоимости. Стеки и очереди.

## 10. Алгоритмы Merge-Sort и Quick-Sort.

Понятие о методе «разделяй и властвуй». Алгоритм Merge-Sort. Слияние двух упорядоченных списков. Оценка сложности. K-wayMerge-Sort для работы во внешней памяти. Сортировка слиянием без использования дополнительной памяти. Общая схема алгоритма Quick-Sort. Два варианта реализации Partition. Примеры неудачного выбора опорных элементов.

## 11. Порядковые статистики. Кучи. Хэширование

Нахождение порядковых статистик с помощью рандомизированной модификации алгоритма Quick-Sort. Линейность матожидания времени работы. Приближенные медианы. Выбор  $k$ -й порядковой статистики за линейное в худшем случае. Деревья со свойствами кучи. Почти полные бинарные деревья: нумерация вершин, навигация. Двоичная куча. Операция просеивания вниз и вверх.

Хеш-функции. Коллизии. Разрешение коллизий методом цепочек, методом последовательных проб и методом двойного хеширования. Гипотеза простого равномерного хеширования, оценка средней длины цепочки. Универсальные семейства хеш-функций, оценка средней длины цепочки.

## 12. Деревья поиска. Система непересекающихся множеств. Задачи RMQ и LCA. Структуры данных для геометрического поиска.

Определение дерева поиска. Вставка и удаление элементов. Inorder-обход дерева. Красно-черные деревья: определение и основные свойства. Реализация операций вставки для красно-черного дерева. Splay-деревья. Операция splay: zig, zig-zig и zig-zag шаги. Реализация операций вставки, удаления, слияния и разделения для splay-деревьев. Декартовы деревья (дучи). Единственность декартова дерева для заданного набора различных ключей и приоритетов. Логарифмическая оценка матожидания высоты дучи. Операции слияния и разделения для дуч. Операции вставки и удаления элементов для дуч.

Задачи RMQ (range minimum query) и LCA (least common ancestor). Сведение от задачи RMQ к задаче LCA, декартово дерево. Алгоритм Таржана для offline-версии задачи LCA. Простейшие алгоритмы для online-версии задачи LCA: полная и разреженная таблицы ответов. Алгоритм Фарах-Колтона-Бендера для задачи  $\pm 1$ -RMQ. Сведение задачи LCA к задаче  $\pm 1$ -RMQ: эйлеров обход дерева.

Location problem, stabbing problem. Деревья интервалов. Сведение системы интервалов к двумерной задаче. Задача поиска точек в коридоре. Priority search tree. Задача поиска точек в прямоугольнике. Дерево отрезков по координате  $X$ , упорядоченные по  $Y$  списки точек в каждой вершине. Сложность  $O(n \log n)$  для построения и  $O(\log^2 n)$  для запроса. Уменьшение времени поиска до  $O(\log n)$ . Задача одновременного поиска в наборе упорядоченных списков. Fractional cascading.

## 13. Поиск кратчайших путей. Минимальные остовные деревья. Минимальные разрезы.

Кратчайшие пути в графе, примеры функции длин. Оценки расстояний и их релаксация. Алгоритмы Форда-Беллмана и Флойда. Алгоритм Дейкстры. Критерий консервативности функции длин дуг в терминах наличия допустимого набора потенциалов. Алгоритм Джонсона для задачи APSP при произвольных длинах дуг. Использование маяков (landmarks) для быстрого поиска кратчайших путей. Алгоритм ALT.

Задача об оптимальном остовном дереве. Хорошие множества, лемма о минимальном ребре в разрезе. Алгоритмы Краскала, Прима и Борушки. Оценки сложности. Задачи о минимальном глобальном разрезе и о минимальном  $s$ - $t$  разрезе, их связь. Стыгивания графа. Алгоритм Штёра-Вагнера.

## 14. Поиск подстрок. Длиннейшие общие подстроки. Приближенный поиск подстрок.

$Z$ -функция: определение и использование в задаче поиска подстроки. Построение  $Z$ -функции за линейное время. Оптимизация поиска подстрок с помощью  $Z$ -функции по памяти. Использование  $Z$ -функции для задачи приближенного поиска подстрок с одной ошибкой за линейное время. Задача множественного поиска подстрок, ожидаемая асимптотика времени работы. Бор для набора слов: определение и способы представления. Префикс-функция на боре. Алгоритм Ахо--Корасик для множественного поиска подстрок.

Задача приближенного поиска подстрок в тексте. Формулировка в терминах расстояний по графу динамического программирования. Алгоритм Ландау-Вишкина: множества достижимых вершин и их границы. База и шаг алгоритма, использование LCP.

## 15. Суффиксные деревья. Суффиксные массивы.

Общая схема алгоритма Укконена для построения сжатого суффиксного дерева за время, линейное по длине строки. Итерации и шаги алгоритма. Классификация шагов. Лемма о возможных переходах между шагами различных типов. Элиминация шагов типа 1: неявные пометки листовых дуг. Элиминация шагов типа 3: досрочное окончание итерации. Оценка количества шагов типа 2. Поиск положений для шагов типа 2: суффиксные ссылки. Прием «скачок по счетчику» для быстрого вычисления суффиксных ссылок. Лемма об изменении вершинной глубины при переходе по суффиксной ссылке.

## 5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Учебная аудитория, оснащенная компьютером и мультимедийным оборудованием (проектор, звуковая система).

## 6. Перечень рекомендуемой литературы

### Основная литература

1. Язык программирования C++ [Текст] = The C++ Programming Language, [учеб. пособие для вузов] / Бьерн Страуструп ; пер. с англ. под ред. Н. Н. Мартынова. - М., БИНОМ, 2017
2. Структуры данных и алгоритмы [Текст] / А. В. Ахо, Д. Э. Хопкрофт, Д. Д. Ульман ; пер. с англ. А. А. Минько. — М. : Вильямс, 2001, 2007. — 384 с.

### Дополнительная литература

1. Структуры данных и алгоритмы Java [Текст] : [учеб. пособие для вузов] / Р. Лафоре ; [пер. с англ. Е. Матвеева] . — 2-е изд. — СПб. : Питер, 2011. — 701 с.

## 7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

1. [http://en.wikipedia.org/wiki/Introduction\\_to\\_Algorithms](http://en.wikipedia.org/wiki/Introduction_to_Algorithms)
2. [http://en.wikipedia.org/wiki/Design\\_Patterns](http://en.wikipedia.org/wiki/Design_Patterns)
3. [http://en.wikipedia.org/wiki/Code\\_Complete](http://en.wikipedia.org/wiki/Code_Complete)
4. <http://misko.hevery.com/2008/11/11/clean-code-talks-dependency-injection/>
5. <http://misko.hevery.com/attachments/Guide-Writing%20Testable%20Code.pdf>
6. [http://en.wikipedia.org/wiki/The\\_Pragmatic\\_Programmer](http://en.wikipedia.org/wiki/The_Pragmatic_Programmer)

## 8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)

Для части занятий потребуется Zoom. Google Drive для доступа к материалам курса. Приветствуется наличие во время занятий смартфонов/ноутбуков для участия в интерактивных упражнениях.

## 9. Методические указания для обучающихся по освоению дисциплины (модуля)

Студент, изучающий дисциплину, должен, с одной стороны, овладеть общим понятийным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике. В результате изучения дисциплины студент должен знать основные определения дисциплины, уметь применять полученные знания для решения различных задач.

Успешное освоение курса требует:

- посещения всех занятий, предусмотренных учебным планом по дисциплине;
- ведения конспекта занятий;
- напряжённой самостоятельной работы студента.

Самостоятельная работа включает в себя:

- чтение рекомендованной литературы;
- проработку учебного материала, подготовку ответов на вопросы, предназначенных для самостоятельного изучения;
- решение задач, предлагаемых студентам на занятиях;
- подготовку к выполнению заданий текущей и промежуточной аттестации.

Показателем владения материалом служит умение без конспекта отвечать на вопросы по темам дисциплины.

Важно добиться понимания изучаемого материала, а не механического его запоминания. При затруднении изучения отдельных тем, вопросов, следует обращаться за консультациями к преподавателю.

Возможен промежуточный контроль знаний студентов в виде решения задач в соответствии с тематикой занятий.

## ПРИЛОЖЕНИЕ

### ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

<b>по направлению:</b>	Прикладные математика и физика
<b>профиль подготовки:</b>	Алгоритмическая биология Физтех-школа Биологической и Медицинской Физики центр образовательных программ Физтех-школы биологической и медицинской физики
<b>курс:</b>	<u>1</u>
<b>квалификация:</b>	магистр
Семестр, формы промежуточной аттестации: 1 (осенний) - Экзамен	
<b>Разработчик:</b>	А.С. Дух, старший преподаватель



## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-2 Имеет представление об актуальных проблемах науки и техники в области своей профессиональной деятельности, способен на научном языке формулировать профессиональные задачи	ОПК-2.1 Имеет представление о современном состоянии исследований в рамках тематической области своей профессиональной деятельности
	ОПК-2.2 Способен оценивать актуальность исследований в области своей профессиональной деятельности и их практическую значимость
ОПК-4 Способен успешно реализовывать решение поставленной задачи, провести анализ результата и представить выводы, применяя знания и навыки в области физико-математических наук и информационно-коммуникационных технологий	ОПК-4.1 Способен применять знания и навыки по использованию информационно-коммуникационных технологий для поиска и изучения научной литературы, применения прикладных программных продуктов
	ОПК-4.2 Способен применять знания в области физико-математических наук для решения поставленной задачи, формулирования выводов и оценки полученных результатов

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Алгоритмы и структуры данных» обучающийся должен:

### знать:

Основные алгоритмы анализа данных, их преимущества и недостатки, а также структуры данных поиска.

### уметь:

Использовать средства языка программирования C++ для разработки надежных и быстро работающих программных систем. Создавать качественный код для реализации алгоритмов анализа данных.

### владеть:

Средствами разработки и тестирования программного кода на языке C++.

## 3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

1. Рандомизированный поиск 2-разрезов на основе fingerprints.cascading.
2. Анализ сложности алгоритма Дейкстры. Использование бинарных и k-ичных куч. Двухнаправленный алгоритм Дейкстры. Системы потенциалов в задаче о кратчайших путях.
3. Поиск подстрок: джокеры типа "?" и "\*". Использование алгоритма Ахо-Корасик. Оценка сложности. Редукция по символу алфавита. Поиск подходящих позиций с использованием сверток.
4. Связь сверток с умножением полиномов. Пара оптимизаций: бинаризация алфавита и сокращение длины текста с помощью двуслойного покрытия.
5. Доказательство линейности времени работы алгоритма Укконена. Суффиксные массивы. Поиск подстрок с помощью суффиксных массивов и бинарного поиска.
6. Построение суффиксного массива за время  $O(n \log n)$  методом маркировки (алгоритм Карпа-Миллера-Розенберга).
7. Задача о длиннейшей общей подстроке. Решение за линейное время с помощью суффиксного дерева и суффиксного массива.
8. Поиск сильно связанных компонент с помощью обхода в глубину. Топологическая сортировка конденсации. 1- и 2-разрезы.
9. Линейное пространство на ребрах графа, его размерность. Подпространство циклов графа, его размерность и базис. Подпространство разрезов графа.
10. Разложение пространства в прямую ортогональную сумму подпространств циклов и разрезов. Генерация случайного равновероятного элемента в пространстве циклов.

#### 4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

Базовые задания:

1. Реализация на основе массива переменного размера и на основе связанного списка.
2. Моделирование очереди с помощью двух стеков. Задача о поддержании динамического максимума в стеке и очереди.
3. Изменяемые (mutable) и неизменяемые (immutable) структуры данных.
4. Структуры данных с хранением истории (persistent). Immutable-стек и immutable-очередь. Проблема множественного будущего при анализе учетных стоимостей в persistent-структурах.
5. Рандомизированный выбор опорного элемента. Сложность Quick-Sort в худшем и среднем случаях.
6. Глубина рекурсии в худшем и среднем случаях. Элиминация хвостовой рекурсии. Задача об оптимальном дереве слияний.
7. Коды Хаффмана. Слияние двух упорядоченных последовательностей различной длины.
8. Теоретико-информационная нижняя оценка. Бинарный поиск "от края" (galloping).
9. Реализация операций вставки, удаления и поиска минимума. Преобразование произвольного массива ключей в кучу (операция Make-Heap), линейность времени работы.
10. Алгоритм сортировки Heap-Sort. k-ичные кучи, зависимость сложности операций от выбора k. Биномиальные (binomial), левацкие (leftlist) и косые (skew) кучи.
11. Построение универсального семейства для целочисленных ключей. Совершенные хеш-функции. Построение совершенной хеш-функции с помощью универсального семейства.
12. Интерфейс множества с ошибками. Фильтр Блюма (Bloomfilter). Оценка вероятности ложноположительного срабатывания. Интерфейс словаря с ошибками. Модификация фильтра Блюма (bloomierfilter).
13. Построение декартового дерева за линейное время при условии предварительной сортировки ключей. В+ деревья: определения и основные свойства.
14. Операции поиска, вставки и удаления для В+ деревьев. Системы непересекающихся множеств.
15. Реализация с использованием леса. Ранги вершин, эвристика ранга.
16. Логарифмическая оценка ранга через количество элементов. Рандомизированная ранговая эвристика.
17. Эвристика сжатия путей. Оценка учетной стоимости операций (без доказательства).

Билет 1

1. Реализация на основе массива переменного размера и на основе связанного списка.
2. Моделирование очереди с помощью двух стеков. Задача о поддержании динамического максимума в стеке и очереди.

Билет 2

1. Построение универсального семейства для целочисленных ключей. Совершенные хеш-функции. Построение совершенной хеш-функции с помощью универсального семейства.
2. Интерфейс множества с ошибками. Фильтр Блюма (Bloomfilter). Оценка вероятности ложноположительного срабатывания. Интерфейс словаря с ошибками. Модификация фильтра Блюма (bloomierfilter).

Критерии оценивания

отлично (10) - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

отлично (9) - выставляется студенту, показавшему свободное оперирование знаниями учебной программы дисциплины, выполнение заданий творческого характера.

отлично (8) - выставляется студенту, показавшему владение программным учебным материалом с наличием несущественных ошибок в действиях, самостоятельно исправляемых учащимся.

хорошо (7) - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускается в ответе или в решении задач некоторые неточности.

хорошо (6) - выставляется студенту если он осознает воспроизведение программного учебного материала, в том числе и различной степени сложности, с несущественными ошибками, затруднения в применении отдельных навыков.

хорошо (5) - выставляется студенту если теоретическое содержание освоено не полностью, некоторые практические навыки сформированы недостаточно, в некоторых случаях были допущены ошибки.

удовлетворительно (4) - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации.

удовлетворительно (3) - выставляется студенту в случае большого количества недочетов и неправильных ответов, а также пассивной работе в ходе занятий, многие учебные задания не выполнены.

неудовлетворительно (2) - выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

неудовлетворительно (1) - выставляется студенту, который не освоил теоретическое и практическое содержание курса, все выполненные учебные задания содержат грубые ошибки.

## **5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности**

Во время проведения экзамена по дисциплине обучающиеся могут пользоваться программой дисциплины. Экзамен проходит путем специального опроса, проводимого в устной форме.