

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

УТВЕРЖДЕНО

**Директор физтех-школы
радиотехники и компьютерных
технологий**

А.В. Дворкович

Рабочая программа дисциплины (модуля)

по дисциплине:	Современные концепции программирования в алгоритмических языках C++ и Java
по направлению:	Информатика и вычислительная техника
профиль подготовки:	Компьютерные технологии и вычислительная техника Физтех-школа Радиотехники и Компьютерных Технологий кафедра микропроцессорных технологий в интеллектуальных системах управления
курс:	4
квалификация:	бакалавр

Семестры, формы промежуточной аттестации:

7 (осенний) - Дифференцированный зачет

8 (весенний) - Экзамен

Аудиторных часов: 60 всего, в том числе:

лекции: 60 час.

семинары: 0 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 45 час.

Подготовка к экзамену: 30 час.

Всего часов: 135, всего зач. ед.: 3

Программу составил: А.Д. Добров, канд. техн. наук

Программа обсуждена на заседании кафедры микропроцессорных технологий в интеллектуальных системах управления 20.04.2020

Аннотация

Курс Современные концепции программирования в алгоритмических языках C++ и Java направлен на освоение студентами фундаментальных знаний в области языков программирования и общей концепции ООП, на примере C++ и Java (синтаксис, семантика реализации на разных архитектурах). В результате освоения программы у студентов сформируются теоретические знания и практические навыки в использовании Объектно-ориентированного подхода к разработке промышленного ПО.

1. Цели и задачи

Цель дисциплины

освоение студентами фундаментальных знаний в области языков программирования (синтаксис, семантика реализации на разных архитектурах), и прежде всего концепций Объектно-ориентированного программирования, разработанных в последние 10-15 лет, на примере двух самых современных языков C++ и Java.

Задачи дисциплины

- формирование общего представления о языках программирования высокого уровня, принципах их построения и реализации;
- формирование базовых знаний о семантике «классического» языка С и основных технологических аспектах разработки программного обеспечения на нем (включая обзор сценарийных средств в виде make-файлов и языков Шелл и Перл)
- формирование теоретических знаний и практических навыков в использовании Объектно-ориентированного подхода на примере C++ и Java (классы, наследование, переопределение операций, ситуационный механизм, шаблоны, программные модели окружения Java).

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
ОПК-3 Способен составлять и оформлять научные и (или) технические (технологические, инновационные) отчеты (публикации, проекты)	ОПК-3.3 Владеет методами визуального и графического представления результатов научной (научно-технической, инновационной технологической) деятельности в виде отчетов, научных публикаций
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности
	ПК-1.2 Способен выдвигать гипотезы, строить математические модели для описания изучаемых явлений и процессов, оценивать качество разработанной модели
	ПК-1.3 Способен применять теоретические и (или) экспериментальные методы исследований к конкретной научной задаче и интерпретировать полученные результаты

ПК-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.1 Знает принципы построения научной работы, методы сбора и анализа полученного материала, способы аргументации
	ПК-2.2 Способен планировать и проводить научные исследования самостоятельно или в качестве члена (руководителя) малого научного коллектива
	ПК-2.3 Способен проводить апробацию результатов научно-исследовательской работы посредством публикации научных статей и участия в конференциях

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- ☐ основополагающие концепции в разработке алгоритмических языков и базовые принципы их реализации для разных архитектур;
- ☐ современные тенденции и направления развития языков программирования;
- ☐ основы базовых технологий по разработке программных систем и соответствующий инструментарий (процедурное и модульное программирование, управление версиями текстовых файлов и программных конфигураций, make-files, сценарийные языки)
- ☐ современные концепции программирования в виде базовых ООП-механизмов и связанной с ними методологии разработки программ;
- ☐ современные модели программирования (библиотека классов Java): полиоконные системы и графика, сеть, параллельное программирование, клиент-сервер взаимодействие, HTML-технология (с включением активных компонентов в браузеры).

уметь:

- ☐ эффективно применять полученные знания о языках программирования в различных областях программистской индустрии;
- ☐ эффективно использовать накопленные знания и навыки для решения задач из различных областей (численное и логическое программирование, системное программирование и инструменты, искусственный интеллект, интерактивные и игровые программы, базы данных и знаний и пр.);
- ☐ на базе полученных знаний быстро адаптироваться при работе в крупных программистских проектах, выполняемых одновременно большим коллективом разработчиков.

владеть:

- ☐ техникой написания программ на языках C, C++ и Java;
- ☐ различными методами организации и разработки программ;
- ☐ технологическими приемами обработки и управления программными комплексами (в том числе сложной структуры).

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	Введение в алгоритмические языки	2			1
2	Семантический базис языков C и C++	4			2
3	Процедуры C и C++	5			2

4	Технология модульного программирования	5			2
5	Системная поддержка технологий в ЮНИКСе	4			2
6	Классы C++ и объектно-ориентированное программирование	4			2
7	Наследование в классах	3			2
8	Перегрузка функций и переопределение операций	3			2
9	Параметризация классов и функций	1			1
10	Механизм исключений	4			3
11	Введение в Джава	2			3
12	От C++ к Джава	3			3
13	Джава – универсальный язык для разработки приложений	2			3
14	Параллельное программирование в Джава	4			4
15	Стандартные классы общего применения	3			3
16	Системные аспекты организации Джава	4			4
17	Сетевые возможности Джава	3			3
18	Оконный и графический интерфейсы Джава	4			3
Итого часов		60			45
Подготовка к экзамену		30 час.			
Общая трудоёмкость		135 час., 3 зач.ед.			

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

Семестр: 7 (Осенний)

1. Введение в алгоритмические языки

Краткая характеристика языков программирования и обзор истории их развития. C++ и JAVA как универсальные современные алгоритмические языки высокого уровня.

2. Семантический базис языков C и C++

Языковые типы данных: скаляры, определяемые типы данных (перечислимый тип, структуры и объединения), массивы и указатели. Типы и операции.

3. Процедуры C и C++

Процедурный механизм, элементы реализации. Доступность имен в процедурных языках. Глобальные и локальные переменные. Разные методы передачи параметров. Указатели на процедуры. Процедурный контекст.

4. Технология модульного программирования

Статические переменные и проблемы линкования. Препроцессор и управление текстовыми модулями. Предварительное объявление и доопределение объектов. h-файлы. Элементы технологии модульного программирования.

5. Системная поддержка технологий в ЮНИКСе

Интерфейс пользователя с системой. Язык-оболочка Shell. Философия и основные конструкции. Версионное управление текстовыми файлами. Системы SCCS и CVS. Конкретные примеры. Управление программными конфигурациями и их версиями. Системы с использованием интерпретатора Make. Синтаксис языка и примеры.

6. Классы C++ и объектно-ориентированное программирование

Изменение парадигмы программирования. Понятие класса. Абстракция в типе. Интерфейс и реализация. Инкапсуляция. Типы и экземпляры объектов (объекты). Элементы реализации на традиционной архитектуре ЭВМ. Объект "this".

7. Наследование в классах

Механизм простого наследования. Дерево (ориентированный граф) наследования. Правило приведения типов. Замещение(экранирование) имен, доступ к закрытому имени. Виртуальные интерфейсные функции. Множественное наследование. Приведение типов при множественном наследовании. Абстрактные класс(интерфейсы). Виртуальные классы.

8. Перегрузка функций и переопределение операций

Перегрузка функций. Управление вызовом. Переопределение операций. Приведение типов как механизм управления при переопределении операций. Унарные, пре- и постфиксные операции. Операции для класса "complex" как пример.

Семестр: 8 (Весенний)

9. Параметризация классов и функций

Проблема получения экземпляров типов периода компиляции. Параметризация класса. Параметризация функций. Формальные правила приоритетов при компиляции конструкции "вызов процедуры".

10. Механизм исключений

Проблематика оператора GOTO. Различные структуры управления и GOTO. Концепция структурного программирования и требования к языку. Реализация локальных и глобальных переходов. Реорганизация стека при далеких переходах (A60, Паскаль). Ситуации и структурный переход в системе Эльбрус. Исключения в C++. Порождение исключения как механизм взаимодействия процедур. Порождение типов и объектов. Их перехват и обработка(try-блок). Не перехваченные исключения. Особенности программирования с использованием исключений. Проблематика ресурсов.

11. Введение в Джава

История возникновения языка Джава. Концепция защищенного (надежного) программирования. Теговая архитектура. Положительные и отрицательные свойства. Варианты преодоления трудностей. Подход Java.

12. От C++ к Джава

Простые типы данных Java. Массивы Java и их отличие от C++. Особенности структур управления. Классы Java. Наследование. Пакеты интерфейсы, реализация. Особенности механизма исключений.

13. Джава – универсальный язык для разработки приложений

"Классовое окружение" Java (Java API). Общий обзор. Native-методы.

14. Параллельное программирование в Джава

Общее понятие процесса. Классификация (независимый процесс, подчиненный, со-процесс). Проблема синхронизации. Примеры организации процессов в разных операционных системах (C&UNIX, Эльбрус). Подход Джава. Порождение процесса и управление им. Концепция синхронизации: synchronized методы, примитивы "wait", "notify", "notify_all". Пример решения задачи "читатель/писатель" на Джава.

15. Стандартные классы общего применения

Строки и массивы. Математические функции. Ввод/вывод (файл, массив, файловые потоки). Программирование Апплетов.

16. Системные аспекты организации Джава

Системные аспекты организации Джава Виртуальная Джава-машина (JVM). Обзор типов данных и операций. Структура класс-файла: методы, константный пул. Реализация типовых конструкций языка на JVM.

17. Сетевые возможности Джава

Сеть. Локальная и глобальная сети. Интернетовский адрес. Протоколы и их общая классификация. Пример работы с оболочками telnet и ftp. Отражение узла Интернет в Джава. Создание сетевых приложений: пример генерации "сокетов" для клиентской и серверной частей, разбор задачи обмена сообщениями между клиентом и сервером.

18. Оконный и графический интерфейсы Джава

Класс AWT и предистория его появления. Иерархия объектов AWT. Структура окна. Обработка событий. Пример мультимедийного приложения.

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

компьютер и мультимедийное оборудование (проектор, звуковая система).

6.Перечень рекомендуемой литературы

Основная литература

1. Язык программирования C [Текст] : [учеб. пособие для вузов] / Б. Керниган, Д. Ритчи ; пер. с англ. и ред. В. Л. Бродового .— 2-е изд., перераб. и доп. — М. : Вильямс, 2006,2007, 2009, 2010, 2012,2013,2015 .— 304 с.
2. Язык программирования Java [Текст]/К. Арнольд, Дж. Гослинг , -СПб., Питер, 1997

Дополнительная литература

1. Язык программирования C++ [Текст] / Б. Страуструп ; пер. с англ. С. Анисимова, М. Кононова ; под ред. Ф. Андреева, А. Ушакова .— Спец. изд. с авт. изменениями и доп. — М. : Бином Пресс, 2008 .— 1104 с.

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

Информационные ресурсы в интернете: сайты с описанием языков программирования C, C+, Java (например, <http://www.java-study.ru/java-uchebnik/2-vvedenie>). Открытые реализации технологических инструментов (GNU, lcc, и т.д.).

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

Мультимедийные технологии. MS PowerPoint, демонстрация презентаций.

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, понятия.

Успешное освоение курса требует напряжённой самостоятельной работы студента. В программе курса приведено минимально необходимое время для работы студента над темой. Самостоятельная работа включает в себя:

- чтение и конспектирование рекомендованной литературы;
- проработку учебного материала (по конспектам, учебной и научной литературе);
- подготовку к экзамену.

Руководство и контроль за самостоятельной работой студента осуществляется в форме индивидуальных консультаций.

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

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Информатика и вычислительная техника
профиль подготовки:	Компьютерные технологии и вычислительная техника Физтех-школа Радиотехники и Компьютерных Технологий кафедра микропроцессорных технологий в интеллектуальных системах управления
курс:	4
квалификация:	бакалавр
Семестры, формы промежуточной аттестации:	
	7 (осенний) - Дифференцированный зачет
	8 (весенний) - Экзамен
Разработчик:	А.Д. Добров, канд. техн. наук

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
ОПК-3 Способен составлять и оформлять научные и (или) технические (технологические, инновационные) отчеты (публикации, проекты)	ОПК-3.3 Владеет методами визуального и графического представления результатов научной (научно-технической, инновационной технологической) деятельности в виде отчетов, научных публикаций
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности
	ПК-1.2 Способен выдвигать гипотезы, строить математические модели для описания изучаемых явлений и процессов, оценивать качество разработанной модели
	ПК-1.3 Способен применять теоретические и (или) экспериментальные методы исследований к конкретной научной задаче и интерпретировать полученные результаты
ПК-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.1 Знает принципы построения научной работы, методы сбора и анализа полученного материала, способы аргументации
	ПК-2.2 Способен планировать и проводить научные исследования самостоятельно или в качестве члена (руководителя) малого научного коллектива
	ПК-2.3 Способен проводить апробацию результатов научно-исследовательской работы посредством публикации научных статей и участия в конференциях

2. Показатели оценивания компетенций

В результате изучения дисциплины «Современные концепции программирования в алгоритмических языках C++ и Java» обучающийся должен:

знать:

- ☐ основополагающие концепции в разработке алгоритмических языков и базовые принципы их реализации для разных архитектур;
- ☐ современные тенденции и направления развития языков программирования;
- ☐ основы базовых технологий по разработке программных систем и соответствующий инструментарий (процедурное и модульное программирование, управление версиями текстовых файлов и программных конфигураций, make-files, сценарийные языки)
- ☐ современные концепции программирования в виде базовых ООП-механизмов и связанной с ними методологии разработки программ;
- ☐ современные модели программирования (библиотека классов Java): полиоконные системы и графика, сеть, параллельное программирование, клиент-сервер взаимодействие, HTML-технология (с включением активных компонентов в браузеры).

уметь:

- ☐ эффективно применять полученные знания о языках программирования в различных областях программистской индустрии;
- ☐ эффективно использовать накопленные знания и навыки для решения задач из различных областей (численное и логическое программирование, системное программирование и инструменты, искусственный интеллект, интерактивные и игровые программы, базы данных и знаний и пр.);
- ☐ на базе полученных знаний быстро адаптироваться при работе в крупных программистских проектах, выполняемых одновременно большим коллективом разработчиков.

владеть:

- ☐ техникой написания программ на языках C, C++ и Java;
- ☐ различными методами организации и разработки программ;
- ☐ технологическими приемами обработки и управления программными комплексами (в том числе сложной структуры).

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

Текущий контроль успеваемости проводится 2 раза в семестр в виде короткой контрольной работы на шестой и одиннадцатой лекции каждого семестра.

Перечень тем проверяемых на контрольных работах:

1-й Семестр:

Контрольная №1

- Принципы построения языка C.
- Семантический базис языка C и процедурный механизм.
- Технология модульного программирования.

Контрольная №2

- Системы управления версиями текстовых файлов.
- Парадигма классов и поддержка их в C++.
- Переопределение операций, шаблоны.

2-й Семестр:

Контрольная №3

- История и принципы разработки системы Java.
- Java как язык программирования.
- Базовые классы Java.

Контрольная №4

- Параллельные процессы Java.
- Многооконная система и графика Java.

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

Перечень контрольных вопросов для сдачи дифференцированного зачета:

1. Языковые структуры данных C: скаляры, определяемые типы (структуры), массивы.
2. Структуры управления в C и C++.
3. Процедурный механизм. Методы передачи параметров. Следствия.

4. Доступность имен в процедурных языках типа С. Глобальные и локальные переменные. Статические переменные. Технология модульного программирования и его основной инструментарий.
5. Классы С++. Изменение парадигмы программирования. Интерфейсы и реализации. Типы и экземпляры объектов (объекты). Управление доступом к объектам и их компонентам.
6. Механизм простого наследования. Приведение типов. Замещение (экранирование) имен. Доступ к "закрытому" имени. Виртуальные функции.
7. Множественное наследование. Приведение типов.
8. Абстрактные классы (интерфейсы). Виртуальные классы.
9. Перегрузка функций. Перегрузка операций. Класс "complex" как пример.

Перечень контрольных вопросов для сдачи экзамена:

1. Языковые структуры данных С: скаляры, определяемые типы (структуры), массивы.
2. Структуры управления в С и С++.
3. Процедурный механизм. Методы передачи параметров. Следствия.
4. Доступность имен в процедурных языках типа С. Глобальные и локальные переменные. Статические переменные. Технология модульного программирования и его основной инструментарий.
5. Классы С++. Изменение парадигмы программирования. Интерфейсы и реализации. Типы и экземпляры объектов (объекты). Управление доступом к объектам и их компонентам.
6. Механизм простого наследования. Приведение типов. Замещение (экранирование) имен. Доступ к "закрытому" имени. Виртуальные функции.
7. Множественное наследование. Приведение типов.
8. Абстрактные классы (интерфейсы). Виртуальные классы.
9. Перегрузка функций. Перегрузка операций. Класс "complex" как пример.
10. Параметризация классов и функций.
11. Проблематика GOTO. Ситуационный механизм.
12. Концепция защищенного (надежного) программирования. Теговая архитектура. Положительные и отрицательные свойства. Варианты преодоления трудностей. Подход Джава.
13. Простые типы данных Джава.
14. Структуры управления Джава и отличия их от С++.
15. Массивы Джава и их отличие от С и С++.
16. Классы Джава. Наследование. Пакеты, интерфейсы, реализация.
17. Особенности ситуационного механизма.
18. "Классовое" окружение Джава. Native-методы. Прикладные пакеты.
19. Ввод/вывод.
19. Параллельные процессы и их синхронизация.
20. Сетевые возможности.
21. Графика Джава (AWT)
22. Апплеты Джава как средство программирования мультимедийных приложений.

Примеры билетов для проведения экзамена:

Билет 1.

1. Языковые структуры данных С: скаляры, определяемые типы (структуры), массивы.
2. Апплеты Джава как средство программирования мультимедийных приложений.

Билет 2.

1. Структуры управления в С и С++.
2. Графика Джава (AWT)

Критерии оценивания

Оценка отлично (10) выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины, проявляющему интерес к данной предметной области, продемонстрировавшему умение уверенно и творчески применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично (9) 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.

Оценка отлично (8) выставляется студенту, показавшему систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений, с некоторыми недочетами.

Оценка хорошо (7) выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но недостаточно грамотно обосновывает полученные результаты.

Оценка хорошо (6) выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности.

Оценка хорошо (5) выставляется студенту, если он в основном знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач достаточно большое количество неточностей.

Оценка 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 освоил основные разделы учебной программы, необходимые для дальнейшего обучения, и может применять полученные знания по образцу в стандартной ситуации.

Оценка удовлетворительно (3) выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, допускающему ошибки в формулировках базовых понятий, нарушения логической последовательности в изложении программного материала, слабо владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и с трудом применяет полученные знания даже в стандартной ситуации.

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

Оценка неудовлетворительно (1) выставляется студенту, который не знает основного содержания учебной программы дисциплины, допускает грубейшие ошибки в формулировках базовых понятий дисциплины и вообще не имеет навыков решения типовых практических задач.

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

Дифференцированный зачет проводится в устной форме.

При проведении зачета обучающемуся предоставляется 20 минут на подготовку. Опрос обучающегося проводится в течение 30 минут.

Во время проведения экзамена и зачета обучающиеся могут пользоваться программой дисциплины.

Экзамен проводится в устной форме.

При проведении устного экзамена обучающемуся предоставляется 1 час на подготовку. Опрос обучающегося по билету на устном экзамене не должен превышать двух астрономических часов.

Во время проведения экзамен обучающиеся могут пользоваться программой дисциплины.