

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

УТВЕРЖДЕНО

**Проректор по учебной работе и
довузовской подготовке**

А.А. Воронов

	Рабочая программа дисциплины (модуля)
по дисциплине:	Информатика в научных исследованиях
по направлению:	Прикладные математика и физика
профиль подготовки:	Пучково-плазменные системы и технологии Физтех-школа Аэрокосмических Технологий кафедра информатики и вычислительной математики
курс:	1
квалификация:	магистр

Семестры, формы промежуточной аттестации:

1 (осенний) - Дифференцированный зачет

2 (весенний) - Дифференцированный зачет

Аудиторных часов: 120 всего, в том числе:

лекции: 0 час.

семинары: 0 час.

лабораторные занятия: 120 час.

Самостоятельная работа: 150 час.

Всего часов: 270, всего зач. ед.: 6

Программу составили:

Т.Н. Дербышева, старший преподаватель

Т.В. Овсянникова, старший преподаватель

Программа обсуждена на заседании кафедры информатики и вычислительной математики 06.02.2020

Аннотация

В курсе учащиеся научатся писать программы на языке Python с применением современных библиотек для решения прикладных задач. Кроме изучения базовых понятий синтаксиса, большое внимание будет уделяться ООП и реализации его принципов как средствами языка, так и сторонних библиотек.

Базовая часть включает работу с функциями, циклами, обзор стандартных коллекций и области видимости данных.

ООП часть опирается на изучение популярных библиотек и написание курсовой работы. В ней учащихся так же знакомят со стандартными модулями, GUI, методикой и практикой тестирования программ

1. Цели и задачи

Цель дисциплины

Научить студентов программировать на языке Python на уровне, достаточном для использования ИКТ в курсе Современные математические пакеты для научных исследований, в исследовательской научной и в последующей профессиональной деятельности.

Задачи дисциплины

1. Обеспечить чёткое понимание студентами основ информатики и ИКТ, включая некоторые области математики (системы счисления, логика, дискретная математика);
2. Обучение студентов принципам создания программных комплексов, выявление особенностей их создания в парадигме объектно-ориентированного программирования;
3. Обучить студентов основным алгоритмам обработки числовой и текстовой информации;"
4. Сформировать у обучающихся навык использования языка программирования Python 3 для решения конкретных прикладных задач;"
5. Сформировать подходы к выполнению исследований студентами в области математического моделирования и численных методов с использованием современных технологий, и программных средств в рамках выпускных работ на степень магистра.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-1 Владеет системой фундаментальных научных знаний в области физико-математических наук	ОПК-1.1 Знает и способен использовать в профессиональной деятельности фундаментальные научные знания в области физико-математических наук
	ОПК-1.2 Способен обобщать и критически оценивать опыт и результаты научных исследований в области профессиональной деятельности
	ОПК-1.3 Понимает междисциплинарные связи в области математики и физики и способен их применять при решении задач профессиональной деятельности
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-2.3 Владеет профессиональной терминологией, используемой в современной научно-технической литературе, обладает навыками устного и письменного изложения результатов научной деятельности в рамках профессиональной коммуникации
ОПК-4 Способен успешно реализовывать решение поставленной задачи, провести	ОПК-4.1 Способен применять знания и навыки по использованию информационно-коммуникационных технологий для поиска и изучения научной литературы, применения прикладных программных продуктов

анализ результата и представить выводы, применяя знания и навыки в области физико-математических наук и информационно-коммуникационных технологий	ОПК-4.3 Способен аргументировано выбирать способ проведения научного исследования
	ОПК-4.2 Способен применять знания в области физико-математических наук для решения поставленной задачи, формулирования выводов и оценки полученных результатов
ОПК-5 Способен и готов к повышению квалификации, профессиональному росту и руководству коллективом в сфере своей профессиональной деятельности, толерантно воспринимая социальные, этнические, конфессиональные и культурные различия	ОПК-5.1 Способен работать в коллективе, толерантно воспринимая социальные, этнические, конфессиональные и культурные различия
	ОПК-5.2 Владеет навыком руководства малым коллективом в сфере своей профессиональной деятельности
	ОПК-5.3 Стремится к получению новых знаний, профессиональному и личностному росту
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-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.1 Способен планировать и проводить научные исследования самостоятельно или в составе научного коллектива
	ПК-2.2 Способен проводить апробацию результатов научно-исследовательской работы посредством публикации научных статей и участия в конференциях
ПК-3 Способен профессионально работать с исследовательским и испытательным оборудованием (приборами и установками, специализированными пакетами прикладных программ) в избранной предметной области	ПК-3.3 Способен оценивать точность полученных экспериментальных (численных) результатов

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны знать:

- Основы теории алгоритмов;
- свойства алгоритмов, проблемы алгоритмической сложности и алгоритмической неразрешимости;
- основы дискретной математики;
- основы алгоритмического языка программирования Python;
- общие характеристики интерпретируемых и компилируемых языков программирования;
- идеологию объектно-ориентированного подхода;
- общие понятия о структурах данных: списки, словари, множества;
- приёмы разработки программ;
- типовые решения, применяемые для создания программ.

уметь:

- Выбирать оптимальные алгоритмы для современных программ;
- разрабатывать полные законченные программы на одном из языков высокого уровня с использованием современных средств написания и отладки программ;
- применять объектно-ориентированный подход для написания программ;
- использовать знания по информатике для приложений в инновационной, конструкторско-технологической и производственно-технологической сферах деятельности;
- работать как на уровне языка командного интерпретатора, так и с использованием графического пользовательского интерфейса.

владеть:

- Навыками самостоятельной работы в среде объектно-ориентированного программирования на языке Python;
- навыками освоения современных архитектур ЭВМ;
- навыками программирования с использованием средств операционной системы для решения исследовательских задач;
- объективной картиной теории и практики объектно-ориентированного программирования.

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	Основы синтаксиса языка. Условные операторы и циклы. Динамическое программирование.			8	8
2	Работа с коллекциями.			12	12
3	Ошибки и исключения			8	8
4	Работа с файлами			8	12
5	Модули и пакеты			8	12
6	ООП			4	11
7	Объекты, типы и классы			12	12
8	Отношение между классами			12	14
9	Методы			8	12
10	Итераторы.			10	12
11	Генераторы.			8	12
12	Технология разработки ПО			10	12
13	Контрольные работы			12	13
Итого часов				120	150
Подготовка к экзамену		0 час.			
Общая трудоёмкость		270 час., 6 зач.ед.			

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

Семестр: 1 (Осенний)

1. Основы синтаксиса языка. Условные операторы и циклы. Динамическое программирование.

Основы синтаксиса языка. Условные операторы и циклы. Динамическое программирование.

2. Работа с коллекциями.

Перебор коллекций. Условные выражения и коллекции. Сравнение коллекций.
Сборщик мусора.

3. Ошибки и исключения

Синтаксические ошибки. Исключения. Генерирование и перехват исключений. Ключевые слова try, except, finally, raise. Поток выполнения при порождении и перехвате исключения.

4. Работа с файлами

Заранее определенные действия по освобождению ресурсов. Конструкция with ... as. Директории."

Запись переменных в файл (модуль pickle).

Чтение и запись в формате csv.

Чтение и запись в формате json.

5. Модули и пакеты

Подключение модулей инструкцией import. Различные синтаксисы import.

Выполнение модуля как скрипта. Пути поиска. dir()

"Компиляция" модулей.

Пакеты. Межпакетные ссылки.

6. ООП

Принципы ООП.

7. Объекты, типы и классы

Определение класса. Создание экземпляра. Конструктор и деструктор.

Время жизни объекта. Инкапсуляция и доступ к свойствам. Приватные переменные.

Полиморфизм. Имитация встроенных типов

Семестр: 2 (Весенний)

8. Отношение между классами

Композиция. Наследование и множественное наследование. Порядок разрешения доступа к методам и полям. «Новые» и «классические» классы. Агрегация. Контейнеры. Итераторы. Ассоциация и слабые ссылки. Метаклассы

9. Методы

Метод. Статический метод. Метод класса. Мультиметоды.

10. Итераторы.

Итераторы - средство упрощения навигации по элементам объекта, который, как правило, представляет собой некоторую коллекцию (список, словарь и т.п.). Поддержка итераторов в языке Python. Итератор - объект переключатель, который для данного объекта выдает следующий элемент, либо бросает исключение, если элементов больше нет. Объекты, элементы которых можно перебирать в цикле for, содержат в себе объект итератор, для того, чтобы его получить необходимо использовать функцию iter, а для извлечения следующего элемента из итератора – функцию next(). Создание собственных итераторов.

11. Генераторы.

Генераторы позволяют значительно упростить работу по конструированию итераторов. Генератор – это функция, которая будучи вызванной в функции next() возвращает следующий объект согласно алгоритму ее работы. Вместо ключевого слова return в генераторе используется yield.

12. Технология разработки ПО

Отладка. Тестирование. Профилирование.

13. Контрольные работы

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

Компьютерный класс с доской, проектором или телевизором, подключенный к сети, установленным необходимым ПО, доступом к учебным материалам и системам перевода текста.

6.Перечень рекомендуемой литературы

Основная литература

1. Программирование на Python 3 : Подробное руководство [Текст] = Programming in Python 3 : [учеб. пособие для вузов] / М. Саммерфилд; пер. с англ. А. Киселева .— СПб : Символ-Плюс, 2015 .— 608 с.

Дополнительная литература

1. Язык программирования PYTHON [Текст] : учеб. пособие для вузов / Р. А. Сузи .— 2 изд., испр. — М. : Интернет-Ун-т Информ. Технологий : БИНОМ. Лаб. знаний, 2007 .— 326 с.

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

Не используются

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

На ПК в компьютерных классах должно быть установлено следующее ПО:

1. Операционная система GNU/Linux.
2. Интерпретатор Python версии не ниже 3.4.
3. Командный интерпретатор Ipython с отладчиком.

На занятиях используются мультимедийные технологии, включая демонстрацию презентаций.

Для контроля и коррекции знаний обучающиеся могут использовать компьютерное тестирование, в том числе на сайте judge.mipt.ru.

В процессе самостоятельной работы обучающихся возможно использование любые среды программирования.

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

Студент, изучающий курс, должен с одной стороны, овладеть общим понятийным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике.

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.

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

Показателем владения материалом служит умение решать задачи. Для формирования умения применять теоретические знания на практике студенту необходимо решать, как можно больше задач. При решении задач каждое действие необходимо аргументировать, ссылаясь на известные теоретические сведения.

При подготовке к лабораторным занятиям необходимо повторять ранее изученные основные понятия.

Теоретические знания даются короткими (15-20 минут) вставками с упором на графическое преподнесение материала и разбор примеров кода. В начале занятия проводится 10-минутный опрос в устной форме по материалам предыдущего занятия.

В задачи для самостоятельной работы включаются методические материалы.

Методические материалы так же пишутся в условиях задач. Фабула и постановка предлагаемых задач включает методическую часть по смежным предметам.

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

Обязательным требованием является решение домашних задач; кроме обязательного прохождения всех тестов задачи необходимо устранить все замечания по алгоритму и стилю оформления кода. Код программы должен содержать комментарии, объясняющие смысл используемых данных. Кроме каждого решения должно дополняться набором проверочных материалов, созданных самим студентом с комментариями по ним.

Промежуточный контроль знаний проводится в виде контрольных работ, на которых студенту предлагается решить несколько задач.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Прикладные математика и физика
профиль подготовки:	Пучково-плазменные системы и технологии Физтех-школа Аэрокосмических Технологий кафедра информатики и вычислительной математики
курс:	<u>1</u>
квалификация:	магистр

Семестры, формы промежуточной аттестации:

- 1 (осенний) - Дифференцированный зачет
- 2 (весенний) - Дифференцированный зачет

Разработчики:

Т.Н. Дербышева, старший преподаватель
Т.В. Овсянникова, старший преподаватель

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-1 Владеет системой фундаментальных научных знаний в области физико-математических наук	ОПК-1.1 Знает и способен использовать в профессиональной деятельности фундаментальные научные знания в области физико-математических наук
	ОПК-1.2 Способен обобщать и критически оценивать опыт и результаты научных исследований в области профессиональной деятельности
	ОПК-1.3 Понимает междисциплинарные связи в области математики и физики и способен их применять при решении задач профессиональной деятельности
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-2.3 Владеет профессиональной терминологией, используемой в современной научно-технической литературе, обладает навыками устного и письменного изложения результатов научной деятельности в рамках профессиональной коммуникации
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-4.3 Способен аргументировано выбирать способ проведения научного исследования
	ОПК-4.2 Способен применять знания в области физико-математических наук для решения поставленной задачи, формулирования выводов и оценки полученных результатов
ОПК-5 Способен и готов к повышению квалификации, профессиональному росту и руководству коллективом в сфере своей профессиональной деятельности, толерантно воспринимая социальные, этнические, конфессиональные и культурные различия	ОПК-5.1 Способен работать в коллективе, толерантно воспринимая социальные, этнические, конфессиональные и культурные различия
	ОПК-5.2 Владеет навыком руководства малым коллективом в сфере своей профессиональной деятельности
	ОПК-5.3 Стремится к получению новых знаний, профессиональному и личностному росту
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-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.1 Способен планировать и проводить научные исследования самостоятельно или в составе научного коллектива
	ПК-2.2 Способен проводить апробацию результатов научно-исследовательской работы посредством публикации научных статей и участия в конференциях
ПК-3 Способен профессионально работать с исследовательским и испытательным оборудованием (приборами и установками, специализированными пакетами прикладных программ) в избранной предметной области	ПК-3.3 Способен оценивать точность полученных экспериментальных (численных) результатов

2. Показатели оценивания компетенций

В результате изучения дисциплины «Информатика в научных исследованиях» обучающийся должен:

знать:

- Основы теории алгоритмов;
- свойства алгоритмов, проблемы алгоритмической сложности и алгоритмической неразрешимости;
- основы дискретной математики;
- основы алгоритмического языка программирования Python;
- общие характеристики интерпретируемых и компилируемых языков программирования;
- идеологию объектно-ориентированного подхода;
- общие понятия о структурах данных: списки, словари, множества;
- приёмы разработки программ;
- типовые решения, применяемые для создания программ.

уметь:

- Выбирать оптимальные алгоритмы для современных программ;
- разрабатывать полные законченные программы на одном из языков высокого уровня с использованием современных средств написания и отладки программ;
- применять объектно-ориентированный подход для написания программ;
- использовать знания по информатике для приложений в инновационной, конструкторско-технологической и производственно-технологической сферах деятельности;
- работать как на уровне языка командного интерпретатора, так и с использованием графического пользовательского интерфейса.

владеть:

- Навыками самостоятельной работы в среде объектно-ориентированного программирования на языке Python;
- навыками освоения современных архитектур ЭВМ;
- навыками программирования с использованием средств операционной системы для решения исследовательских задач;
- объективной картиной теории и практики объектно-ориентированного программирования.

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

Для текущего контроля используются мини-опросы в начале занятий по пройденным темам.

3. Перечень типовых контрольных заданий, используемых для оценки знаний, умений, навыков

Примерный перечень контрольных вопросов по теории за 9-й семестр:

1. Основы архитектуры компьютера. Принципы фон Неймана.
2. Отличие интерпретируемых и компилируемых языков.
3. Концепция присваивания в Python
4. Обмен двух переменных значениями.
5. Кортежи и их использование.
6. Цикл while. Инструкции управления циклом.
7. Позиционные системы счисления
8. Однопроходные алгоритмы: подсчёт, сумма, произведение.
9. Оператор if. Каскадная условная конструкция elif.
10. Логические операции в Python.
11. Основы алгебры логики
12. Однопроходные алгоритмы: поиск числа в потоке, максимум.
13. Тест простоты числа.
14. Разложение числа на множители.
15. Тип str. Длина строки len(s). Неизменяемость строки.
16. Срезы строк.
17. Методы строк find, count, replace, startswith, endswith.
18. Наивный поиск подстроки в строке.
19. Ссылочная модель данных в Python. Операторы == и is. Копирование объектов.
20. Алгоритм обращения массива.
21. Алгоритм циклического сдвига в массиве.
22. Срезы списков. Присваивание в срез. Методы списка.
23. Список строк. Методы split и join для строки.
24. Цикл for и его особенности в Python.
25. List comprehensions: генерация списков.
26. Двумерные массивы (списки списков). Вложенная генерация.
27. Полиморфизм в Python. Duck typing.
28. Именованные параметры.
29. Поиск корня функции методом бисекции.
30. Поиск значения в упорядоченном массиве методом бисекции.
31. Сортировка обезьяны.
32. Сортировка вставками.
33. Сортировка выбором.
34. Сортировка методом пузырька.
35. Сортировка дурака
36. Сортировка подсчётом.
37. Поразрядная сортировка.
38. Прагматическая сортировка TimSort.
39. Рекурсия. Прямой и обратный ход рекурсии.
40. Проблема алгоритмической сложности задачи.
41. Ханойские башни.
42. Генерация всех перестановок (рекурсивная)
43. Одномерное динамическое программирование.
44. Двумерное динамическое программирование
45. Рекурсия с кешированием на примере факториала.
46. Рекурсивные сортировки. Быстрая сортировка. Сортировка слиянием.
47. Пирамида (куча). Пирамидальная сортировка.
48. Устойчивость сортировок.

49. Тип set. Множества и работа с ними.
50. Тип dict. Словарь (ассоциативный массив) и операции с ним.
51. Dict comprehensions: генерация множеств и словарей.
52. Частотный анализ для строк.
53. Генераторы, yield.

Примерный перечень контрольных вопросов по теории за 10-й семестр:

1. Классы в Python. Перегрузка операторов.
2. Исключения в Python. Генерирование и перехват исключений.
3. Списки: односвязный, двусвязный, кольцо.
4. Стек. Дек.
5. Очередь.
6. Очередь с приоритетами. Пирамида (куча).
7. Очередь событий графического приложения.
8. Хеш-функция. Хеширование.
9. Открытая хеш-таблица.
10. Закрытая хеш-таблица.
11. Проблема удаления из закрытой хеш-таблицы. Перехеширование.
12. Взвешенный граф.
13. Расстояние между двумя вершинами.
14. Графы и способы их представления: список рёбер, матрица смежности, списки смежности
15. Определение дерева.
16. Поиск в глубину.
17. Связность неориентированных графов: выделение компонент связности.
18. Поиск в ширину.
19. Алгоритм Дейкстры.
20. Восстановление кратчайшего пути.
21. Построение гамильтонова цикла.
22. Эйлеров цикл. Эйлеров путь.
23. Минимальное остовное дерево. Алгоритм Прима.
24. Задача о коммивояжере
25. Ографы.
26. Топологическая сортировка.
27. Двоичное дерево поиска.
28. Декартово дерево («дуча»).
29. Балансировка деревьев. AVL-дерево. Красно-чёрное дерево.
30. Проверка равенства строк. Простой и вероятностный алгоритмы.
31. Вычисление расстояния Левенштейна.
32. Поиск подстроки в строке.
33. Алгоритм Рабина-Карпа
34. Конечный автомат для поиска подстроки «abcd», «ababc».

На дифференцированном зачёте предлагается ответить на два-три вопроса по теории и решить одну короткую алгоритмическую задачу на бумаге без использования компьютера.

Пример задания на устном зачёте:

1. Сортировка обезьяны.
2. Рекурсия. Прямой и обратный ход рекурсии.
3. Тип set. Множества и работа с ними.
4. Задача: реализовать слияние двух отсортированных списков.

4. Критерии оценивания

Промежуточная аттестация по дисциплине «Информатика в научных исследованиях» осуществляется в форме дифференцированного зачета.

Дифференцированный зачёт принимает лектор в устной форме с учётом оценки по контекстам и оценки по лабораторному практикуму. Устный ответ практически исключает списывание, показывает владение базовой терминологией предмета, умение говорить на языке информатики, а также позволяет проверить знание сложных алгоритмов, которые долго программируются, но могут быть относительно легко устно объяснены.

Оценка по десятибалльной шкале за работу на лабораторном практикуме выставляется преподавателем практикума исходя из количества и качества выполненных практических работ за семестр. Оценка за выполнение контекстов выставляется автоматически исходя из суммарного рейтинга обучающегося в системе Ejudge и также нормируется к десятибалльной шкале.

За каждый контрольный вопрос из контрольного задания студент получает от 0 до максимального балла в зависимости от полноты представленного ответа (решения). Критерии проставления баллов утверждаются на заседании учебно-методической комиссии кафедры. Процент суммарно набранных баллов от максимально возможного количества определяет оценку за теоретические знания по каждому контрольному заданию:

Оценка	Набранные баллы
отлично (10)	более 88%
отлично (9)	от 78% до 88% включительно
отлично (8)	от 68% до 78% включительно
хорошо (7)	от 58% до 68% включительно
хорошо (6)	от 48% до 58% включительно
хорошо (5)	от 38% до 48% включительно
удовлетворительно (4)	от 28% до 38% включительно
удовлетворительно (3)	от 18% до 28% включительно
неудовлетворительно (2)	от 08% до 18% включительно
неудовлетворительно (1)	не более 08%

Итоговая оценка за зачёт не может отличаться от среднего арифметического оценок по контекстам и по практическим лабораторным работам более чем на три балла.

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

Время проведения зачёта составляет 30 минут на одного обучающегося.

Во время подготовки к ответу обучающиеся не могут пользоваться литературой, печатными материалами, рукописными записями, а также электронными средствами (сотовыми телефонами, планшетами, умными часами и т.п.).