

*Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования «Северо-Осетинский государственный университет  
имени Коста Левановича Хетагурова»*



## **РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ «Современные языки программирования»**

Направление подготовки 01.03.02 Прикладная математика и информатика

Профиль: Математическое моделирование и вычислительная математика

**Форма обучения – очная**

Владикавказ, 2019

Программа составлена в соответствии с Федеральным государственным образовательным стандартом высшего образования по направлению подготовки 01.03.02 Прикладная математика и информатика, утвержденным приказом Министерства образования и науки Российской Федерации от 10.01.2018 г. № 9, учебным планом подготовки бакалавров по направлению подготовки 01.03.02 Прикладная математика и информатика, утвержденным Ученым советом ФГБОУ ВО «СОГУ» от 28.05.2019 г. № 10.

Составитель: старший преподаватель Макаренко М.Д.

Рабочая программа обсуждена и утверждена на заседании кафедры алгебры и геометрии.  
(протокол №7 от 14.03.2019)

Одобрена советом факультета математики и информационных технологий  
(протокол №5 от 29.03.2019)

## 1. Структура и общая трудоемкость дисциплины

Общая трудоемкость дисциплины составляет 4 зачётные единицы. (144 час.).

	Очная Форма обучения
Курс	3
Семестр	5/6
Лекции	34
Практические занятия	-
Лабораторные занятия	34/64
Консультации	-/-
Итого аудиторных занятий	68/64
Самостоятельная работа	4/8
Курсовая работа	-
Зачет	+/-
Экзамен	-/-
Общее количество часов	144 час.

## 2. Цели освоения дисциплины

Целью курса является изучение отличительных конструкций языка Python, так как с одной стороны он излагается студентам уже умеющим программировать, с другой стороны является одним из самых простых, популярных и быстро развивающихся языков программирования, который используется при решении широкого круга задач – от анализа данных до сетевых технологий

## 3. Место дисциплины в структуре ОПОП:

Дисциплина «Современные языки программирования» относится к дисциплинам Блок 1. Дисциплины (модули). Обязательная часть. Б1.О.21.

Для изучения дисциплины необходимы знания, полученные обучающимися в рамках «Языки программирования и методы трансляций», «Информатика» а также в результате освоения дисциплин: «Алгебра и геометрия» и «Математический анализ».

Приступая к изучению дисциплины «Программирование на языке Python», студент должен иметь представление об алгоритмах, структурах данных, парадигмах программирования. Иметь опыт разработки программного обеспечения, проектной работы.

## 4. Требования к результатам освоения дисциплины

Компетенции обучающегося, формируемые в результате освоения дисциплины.

В результате изучения дисциплины обучающийся должен обладать следующими компетенциями (результатами освоения образовательной программы):

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

УК-3 -Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде;

ОПК-2 -Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач.

Взаимосвязь планируемых результатов обучения по дисциплине с формируемыми компетенциями ОПОП:

Компетенции		Планируемые результаты обучения, соответствующие формируемым компетенциям ОПОП		
Код	Формулировка			
		Знать:	Уметь	Владеть:
УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	— основные методы критического анализа; — методологию системного подхода	выявлять проблемные ситуации, используя методы анализа, синтеза и абстрактного мышления;	технологиями выхода из проблемных ситуаций, навыками выработки стратегии действий; — навыками критического анализа.
УК-3	Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде	принципы формирования концепции проекта в рамках обозначенной проблемы; — основные требования, предъявляемые к проектной работе и критерии оценки результатов проектной деятельности;	— осуществлять поиск решений проблемных ситуаций на основе действий, эксперимента и опыта	– навыками составления плана графика реализации проекта в целом и плана-контроля его выполнения; – навыками конструктивного преодоления возникающих разногласий и конфликтов.
ОПК-2	Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач	основные закономерности развития науки и техники, основные принципы и положения философии технических знаний	применять методологию научных исследований и методологию научного творчества	навыками публичной речи, аргументации, ведения дискуссии

При проведении учебных занятий обеспечивается развитие у обучающихся навыков командной работы, межличностной коммуникации, принятия решений, лидерских качеств (включая при необходимости проведение интерактивных лекций, групповых дискуссий, ролевых игр, тренингов, анализ ситуаций и имитационных моделей, преподавание дисциплин (модулей) в форме курсов, составленных на основе результатов научных исследований, в том числе с учетом региональных особенностей профессиональной деятельности выпускников и потребностей работодателей).

## 5. Содержание и учебно-методическая карта дисциплины

Таблица 5.1

Номер недели	Наименование тем (вопросов), изучаемых по данной дисциплине	Занятия			Самостоятельная работа студентов		Формы контроля	Баллы		Литература
		л	пр	лаб	Содержание	Часы		min	max	
1-2	История языков и парадигм программирования. Место языка Python в многообразии современных технологий.			4			Формирование списка ст.			[1-8]
3-14	Списки, срезы, кортежи, словари, итераторы, типы аргументов функций, делегаты, классы, библиотеки			24			DataCamp			[1-8]
15-17	Разработка проекта			6	Подготовка отчетов и презентаций	4				[1-8]
	<b>ИТОГО</b>			<b>34</b>		<b>4</b>		<b>0</b>	<b>100</b>	
1-3	Предобработка данных, дерево решений			6			DataCamp			[1-8]
4-6	Классификаторы, метод k-ближайших соседей			6			DataCamp			[1-8]
7-12	Линейная регрессия, нейронные сети			12			DataCamp			[1-8]
13-15	Представление работ по участию в соревнованиях по машинному обучению			40	Подготовка отчетов и презентаций	8	Защита проектов			[1-8]
	<b>ИТОГО</b>			<b>64</b>		<b>8</b>		<b>0</b>	<b>100</b>	

**Примечания:**

- Все виды учебной работы могут проводиться дистанционно на основании локальных нормативных актов.
- В целях реализации индивидуального подхода к обучению студентов, осуществляющих учебный процесс по индивидуальной траектории в рамках индивидуального рабочего плана, изучение данной дисциплины может осуществляться через индивидуальные консультации преподавателя очно, в часы консультаций, по электронной почте и с использованием платформ дистанционного обучения.

## 6. Образовательные технологии

В соответствии с государственными образовательными стандартами высшего образования реализация учебного процесса должна предусматривать проведение занятий в интерактивных и активных формах. Внедрение этих форм обучения – одно из важнейших направлений совершенствования подготовки студентов в современном вузе. Цель – повышение эффективности образовательного процесса, достижение всеми обучающимися высоких результатов обучения.

Интерактивные формы проведения занятий предполагают обучение в сотрудничестве. Все участники образовательного процесса (преподаватель и студенты) взаимодействуют друг с другом, обмениваются информацией, совместно решают проблемы, моделируют ситуации. Суть использования активных и интерактивных форм проведения состоит в погружении студентов в реальную атмосферу делового сотрудничества по разрешению проблем, оптимальную для выработки навыков и качеств будущего специалиста.

Для решения воспитательных и учебных задач преподавателем могут быть использованы следующие интерактивные формы обучения.

**Традиционные лекции и практические (семинарские) занятия** с использованием современных интерактивных технологий.

**Лекция-диалог** – содержание подается через серию вопросов, на которые студент должен отвечать непосредственно в ходе лекции.

**Онлайн-семинар** – разновидность веб-конференции, проведение онлайн-встреч или презентаций через Интернет в режиме реального времени. Каждый из участников находится у своего компьютера (средства связи), а связь между ними поддерживается через Интернет посредством загружаемого приложения, установленного на компьютере каждого участника.

**Видеоконференция** – сеанс видеоконференцсвязи (ВКС) – это технология интерактивного взаимодействия двух и более участников образовательного процесса для обмена информацией в реальном режиме времени.

**Видео-лекция** – снятая на камеру сокращенная лекция, дополненная фотографиями и схемами, иллюстрирующая подаваемый в лекции материал.

**Технология электронного обучения** (реализуется при помощи электронной образовательной среды СОГУ при использовании ресурсов ЭБС, при проведении автоматизированного тестирования и т. д.).

**Творческое задание** составляет содержание (основу) любой интерактивной формы проведения занятия. Выполнение творческих заданий требует от студента воспроизведения полученной ранее информации в форме, определяемой преподавателем и требующей творческого подхода: 1) подборка примеров из практики; 2) подборка материала по определенной проблеме;

**Публичная презентация проекта** - самый эффективный способ донесения важной информации при публичных выступлениях. Слайд-презентации позволяют эффектно и наглядно представить содержание, выделить и проиллюстрировать сообщение.

**Интерактивная лекция** представляет собой выступление преподавателя перед аудиторией студентов с применением следующих интерактивных форм обучения: 1. управляемая дискуссия или беседа; 2. демонстрация слайдов или учебных фильмов; 3. мозговой штурм; 4. мотивационная речь и др.

**Разработка проекта** позволяет участникам мысленно выйти за пределы аудитории и составить проект своих действий по обсуждаемому вопросу. Участники могут обратиться за консультацией, дополнительной литературой в специализированные учреждения, библиотеки и т.д.

**Проблемное обучение** - поиск ответов на вопросы по теме.

## 7. Учебно-методическое обеспечение самостоятельной работы

Самостоятельная работа обучающихся является одним из видов учебных занятий. Самостоятельная работа проводится с целью:

- систематизации и закрепления полученных теоретических знаний и практических умений обучающихся студентов;
- углубления и расширения теоретических знаний;
- формирования умений самостоятельно находить необходимую литературу и анализировать ее;
- формирования самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развития исследовательских умений.

К видам самостоятельной работы при изучении данной дисциплины относятся: работа с мооc курсами, написание отчетов о проекте, подготовка доклада по защите проекта, выполнение домашних заданий в GoogleCollab и т.д.

Список MOOK для выбора студента:

1. "Поколение Python" (BEEGEEK) - слишком простой <https://stepik.org/course/58852/syllabus>
2. "Программирование на Python" (Институт БИОИНФОРМАТИКИ ИТМО) - простой <https://stepik.org/course/67/syllabus>
3. "Python: основы и применение" (Институт БИОИНФОРМАТИКИ ИТМО) - продвинутый <https://stepik.org/lesson/24459/step/1?unit=6764>
4. "Adaptive Python" (JetBrains, англ.) - сложность задач подбирается под ваш уровень <https://stepik.org/course/568>
5. "Основы программирования на Python" (ВШЭ) - норм <https://www.coursera.org/learn/python-osnovy-programm..>
6. "Погружение в Python" (МФТИ) - норм <https://www.coursera.org/learn/diving-in-python#sylla..>
7. "Основы Python и анализа данных" (Yandex практикум, частично бесплатный) - норм <https://praktikum.yandex.ru/profile/data-analyst/>

Список самостоятельных работ:

№	ССЫЛКА
1	<a href="https://drive.google.com/file/d/1Ka4J5YhTBr20gjLwu_u8oWUcioWkLD4B/view?usp=sharing">https://drive.google.com/file/d/1Ka4J5YhTBr20gjLwu_u8oWUcioWkLD4B/view?usp=sharing</a>
2	<a href="https://drive.google.com/file/d/1eyh0ptQ4j2VEMvjSMYMFojoNAaKf9adz/view?usp=sharing">https://drive.google.com/file/d/1eyh0ptQ4j2VEMvjSMYMFojoNAaKf9adz/view?usp=sharing</a>
3	<a href="https://drive.google.com/file/d/1E5FNDhcBwA1S-J8gj9NKHcltF2gwsdJE/view?usp=sharing">https://drive.google.com/file/d/1E5FNDhcBwA1S-J8gj9NKHcltF2gwsdJE/view?usp=sharing</a>
4	<a href="https://drive.google.com/file/d/1sh7E4dn1bWXjUuyR2zNStrKJ225dBDV6/view?usp=sharing">https://drive.google.com/file/d/1sh7E4dn1bWXjUuyR2zNStrKJ225dBDV6/view?usp=sharing</a>
5	<a href="https://drive.google.com/file/d/1o5nklj1qJ_wDXW2ENSvFcsgS5aj2J0Um/view?usp=sharing">https://drive.google.com/file/d/1o5nklj1qJ_wDXW2ENSvFcsgS5aj2J0Um/view?usp=sharing</a>
6	<a href="https://drive.google.com/file/d/15TeX7wXzruAXQJnwLFWdgnHX2HBWimrT/view?usp=sharing">https://drive.google.com/file/d/15TeX7wXzruAXQJnwLFWdgnHX2HBWimrT/view?usp=sharing</a>
7	<a href="https://drive.google.com/file/d/1fSBCUG6WLBgdbnIj5CdqZQTbalCQER6/view?usp=sharing">https://drive.google.com/file/d/1fSBCUG6WLBgdbnIj5CdqZQTbalCQER6/view?usp=sharing</a>

## **8. Оценочные средства для текущего контроля успеваемости, рубежной аттестации и промежуточной аттестации по итогам освоения дисциплины**

Рабочая программа предусматривает проведение лекционных и практических занятий, а также следующие виды работ: самостоятельную работу студентов в MOOK, коллоквиум по языку Python, выполнение домашних работ по машинному обучению, защита проектов по любому конкурсу по машинному обучению.

*Текущий контроль* – происходит через классы или другие инструменты MOOK.

*Рубежный контроль* осуществляется по окончании изучения двух модулей. Коллоквиум по завершению модуля по языку Python и защита проекта по завершению модуля введение в машинное обучение.

Рубежный контроль проводится с целью определения качества усвоения материала учебного модуля в целом.

### **1) Задачи коллоквиума модуля Язык Python:**



1	<p>Напишите функцию <b>whipping_waterfall()</b>, которая принимает произвольное количество кортежей из двух положительных целых чисел, преобразует их так: находит разность между большим и меньшим, из каждой четной цифры делает нечетную (+1), затем сдвигает циклически влево на количество позиций, которое записано в именованном параметре <b>shift</b>, имеющем значение по умолчанию 0.</p> <p>Функция возвращает список полученных значений.</p> <p>Если в каком-нибудь кортеже есть одинаковые числа, то возбуждается исключение <b>SameValuesError</b> с сообщением <i>Same numbers in tuple</i>.</p> <p>Если позиционные аргументы не переданы, возбуждается стандартное исключение <b>ValueError</b>, в качестве сообщения указывается строка <i>Empty input data</i>.</p> <p>Если оба числа в кортеже однозначные, то возбуждается исключение <b>OneDigitError</b>, его сообщение: <i>Too small value</i>.</p> <p><b>Пример 1</b></p> <table border="0"> <tr> <td><b>Ввод</b></td><td><b>Вывод</b></td></tr> <tr> <td> <pre>args = [(1984, 42), (12345, 516), (2020, 19), (151, 8574)] shift = 2 print(whipping_waterfall(*args, shift=shift))</pre> </td><td> <pre>[5319, 93911, 1131, 3395]</pre> </td></tr> </table> <p><b>Пример 2</b></p> <table border="0"> <tr> <td><b>Ввод</b></td><td><b>Вывод</b></td></tr> <tr> <td> <pre>args = [] shift = 2 print(whipping_waterfall(*args, shift=shift))</pre> </td><td> <pre>ValueError raised with "Empty input data"</pre> </td></tr> </table> <p><b>Пример 3</b></p> <table border="0"> <tr> <td><b>Ввод</b></td><td><b>Вывод</b></td></tr> <tr> <td> <pre>args = [(1999, 142), (1, 5), (20, 19), (11, 85)]</pre> <pre>print(whipping_waterfall(*args))</pre> </td><td> <pre>OneDigitError raised with "Too small value"</pre> </td></tr> </table> <p><b>Примечания</b></p> <p>Обработка исключений производится проверяющей системой.</p> <p>Она же и генерирует сообщения вида <b>Error raised with message "String"</b></p> <p>Поскольку один и тот же набор данных может порождать разные исключения, стоит придерживаться порядка проверки, указанного в условии.</p>	<b>Ввод</b>	<b>Вывод</b>	<pre>args = [(1984, 42), (12345, 516), (2020, 19), (151, 8574)] shift = 2 print(whipping_waterfall(*args, shift=shift))</pre>	<pre>[5319, 93911, 1131, 3395]</pre>	<b>Ввод</b>	<b>Вывод</b>	<pre>args = [] shift = 2 print(whipping_waterfall(*args, shift=shift))</pre>	<pre>ValueError raised with "Empty input data"</pre>	<b>Ввод</b>	<b>Вывод</b>	<pre>args = [(1999, 142), (1, 5), (20, 19), (11, 85)]</pre> <pre>print(whipping_waterfall(*args))</pre>	<pre>OneDigitError raised with "Too small value"</pre>
<b>Ввод</b>	<b>Вывод</b>												
<pre>args = [(1984, 42), (12345, 516), (2020, 19), (151, 8574)] shift = 2 print(whipping_waterfall(*args, shift=shift))</pre>	<pre>[5319, 93911, 1131, 3395]</pre>												
<b>Ввод</b>	<b>Вывод</b>												
<pre>args = [] shift = 2 print(whipping_waterfall(*args, shift=shift))</pre>	<pre>ValueError raised with "Empty input data"</pre>												
<b>Ввод</b>	<b>Вывод</b>												
<pre>args = [(1999, 142), (1, 5), (20, 19), (11, 85)]</pre> <pre>print(whipping_waterfall(*args))</pre>	<pre>OneDigitError raised with "Too small value"</pre>												
2	<p>Напишите функцию <b>chocolate_shop()</b>, которая принимает неограниченное число кортежей из двух строк-слов. В каждой строке буквы заменяются индексом под которым она стоит в слове, если его неповторяющиеся буквы отсортировать по алфавиту. Полученные числа переставляются в кортеже по возрастанию, если их рассматривать как строки, а затем из второго вычитается первое. Волшебство!</p>												

	<p>Кроме позиционных аргументов функция принимает именованный параметр <b>length</b>, по умолчанию 2, – кортежи, в которых есть слова короче этого числа, не рассматриваются.</p> <p>Функция возвращает список полученных чисел.</p> <p>Если после удаления кортежей с короткими словами ничего не осталось, возбуждается исключение <b>EmptyInputError</b>, в качестве сообщения указывается строка <i>No input data</i>.</p> <p>Если параметр <i>length</i> имеет значение меньше 2, возбуждается стандартное исключение <b>AttributeError</b> с сообщением <i>Invalid value</i>.</p> <p>Если в итоговом списке есть отрицательные значения, то возбуждается исключение <b>NegativeError</b>, его сообщение: <i>Negative values in result</i>.</p> <p><b>Пример 1</b></p> <p><b>Ввод</b></p> <pre>args = [('shop', 'plant'), ('choco', 'beans'), ('magic', 'candy'), ('Willy', 'Wonka')]</pre> <pre>print(chocolate_shop(*args))</pre> <p><b>Пример 2</b></p> <table border="0"> <tr> <td><b>Ввод</b></td><td><b>Вывод</b></td></tr> <tr> <td> <pre>args = [('chocolate', 'sweet'), ('valley', 'river')] length = 8 print(chocolate_shop(*args, length=length))</pre> </td><td> <pre>EmptyInputError raised with input data"</pre> </td></tr> </table> <p><b>Пример 3</b></p> <table border="0"> <tr> <td><b>Ввод</b></td><td><b>Вывод</b></td></tr> <tr> <td> <pre>args = [('attractor', 'zoom'), ('valley', 'river')] length = 3 print(chocolate_shop(*args, length=length))</pre> </td><td> <pre>NegativeError raised with values in result"</pre> </td></tr> </table> <p><b>Примечания</b></p> <p>Обработка исключений производится проверяющей системой.</p> <p>Она же и генерирует сообщения вида <b>Error raised with message "String"</b></p> <p>Поскольку один и тот же набор данных может порождать разные исключения, стоит придерживаться порядка проверки, указанного в условии.</p>	<b>Ввод</b>	<b>Вывод</b>	<pre>args = [('chocolate', 'sweet'), ('valley', 'river')] length = 8 print(chocolate_shop(*args, length=length))</pre>	<pre>EmptyInputError raised with input data"</pre>	<b>Ввод</b>	<b>Вывод</b>	<pre>args = [('attractor', 'zoom'), ('valley', 'river')] length = 3 print(chocolate_shop(*args, length=length))</pre>	<pre>NegativeError raised with values in result"</pre>
<b>Ввод</b>	<b>Вывод</b>								
<pre>args = [('chocolate', 'sweet'), ('valley', 'river')] length = 8 print(chocolate_shop(*args, length=length))</pre>	<pre>EmptyInputError raised with input data"</pre>								
<b>Ввод</b>	<b>Вывод</b>								
<pre>args = [('attractor', 'zoom'), ('valley', 'river')] length = 3 print(chocolate_shop(*args, length=length))</pre>	<pre>NegativeError raised with values in result"</pre>								
3	<p><b>Пустота в желудке</b></p> <p>Напишите программу, которая попытает найти еду. Нужно определить, можно ли из любых символов первой строки собрать вторую строку без учета регистра.</p> <p><b>Формат ввода</b></p> <p>В файле <b>stomach.txt</b> находятся две строки.</p> <p><b>Формат вывода</b></p> <p>Если собрать вторую строку можно, то в файл <b>food.txt</b> записать кортеж индексов символов из первой строки; если есть несколько подходящих символов, то брать индекс</p>								

	<p>того, что ближе к началу строки. Если во второй строке символы повторяются, то каждый раз из первой строки берется индекс ближайшего к началу.</p> <p>Если собрать нельзя, то записать <b>None</b>.</p> <p><b>Пример 1</b></p> <table> <tr> <td><b>Ввод</b></td><td><b>Вывод</b></td></tr> <tr> <td>He really wanted something more satisfying and delicious candy</td><td>(51, 5, 12, 15, 8)</td></tr> </table> <p><b>Пример 2</b></p> <table> <tr> <td><b>Ввод</b></td><td><b>Вывод</b></td></tr> <tr> <td>But more than anything, he wanted to... chocolate.</td><td>None</td></tr> </table> <p><b>Примечания</b></p> <p>В задаче ничего не вводится с клавиатуры и ничего не выводится в консоль, все действия производятся с файлами. В примерах показано их содержимое.</p>	<b>Ввод</b>	<b>Вывод</b>	He really wanted something more satisfying and delicious candy	(51, 5, 12, 15, 8)	<b>Ввод</b>	<b>Вывод</b>	But more than anything, he wanted to... chocolate.	None				
<b>Ввод</b>	<b>Вывод</b>												
He really wanted something more satisfying and delicious candy	(51, 5, 12, 15, 8)												
<b>Ввод</b>	<b>Вывод</b>												
But more than anything, he wanted to... chocolate.	None												
4	<p>Напишите программу, которая поможет определиться и, наконец, выбрать нужный рост.</p> <p>По первой последовательности нужно определить минимум и максимум (для простоты назовем их <math>f_{min}</math> и <math>f_{max}</math>) и все числа второй последовательности, которые не попадают в</p> $\left[ \frac{f_{min} + f_{max}}{3}, +\infty \right)$ <p>диапазон.</p> <p><b>Формат ввода</b></p> <p>В первой строке задается количество наборов данных (натуральное число <math>N</math>), которые требуется проанализировать.</p> <p>В следующих <math>2N</math> строках записанные последовательности натуральных чисел разделенные пробелами.</p> <p>В строке <math>2i</math> записана первая часть <math>i</math>-го набора, а в строке <math>2i+1</math> - вторая.</p> <p><b>Формат вывода</b></p> <p>Требуется вывести <math>2N</math> строк, содержащих для каждого из наборов:</p> <ul style="list-style-type: none"> <li>- Минимум и максимум для первой части набора;</li> <li>- Все числа второй части набора, которые попадают под заданное условие.</li> </ul> <p><b>Пример 1</b></p> <table> <tr> <td><b>Ввод</b></td><td><b>Вывод</b></td></tr> <tr> <td>1</td><td>2 8</td></tr> <tr> <td>7 2 8</td><td>1 2</td></tr> <tr> <td>1 2 4 8 16 32</td><td></td></tr> </table> <p><b>Пример 2</b></p> <table> <tr> <td><b>Ввод</b></td><td><b>Вывод</b></td></tr> <tr> <td>2</td><td>8 45</td></tr> </table>	<b>Ввод</b>	<b>Вывод</b>	1	2 8	7 2 8	1 2	1 2 4 8 16 32		<b>Ввод</b>	<b>Вывод</b>	2	8 45
<b>Ввод</b>	<b>Вывод</b>												
1	2 8												
7 2 8	1 2												
1 2 4 8 16 32													
<b>Ввод</b>	<b>Вывод</b>												
2	8 45												

15 8 32 45	1 3 9
1 3 9 27 81	2 15
3 15 4 8 2	1 4
1 4 7 10 13 16 19	
<b>Примечания</b>  Порядок чисел в строках не имеет значения.  Порядок строк — важен.	

Оценка	Критерий оценки коллоквиума
25	Все требования по написанию программы выполнены, на все вопросы студент отвечает быстро и самостоятельно
20 -24	Все требования по написанию программы выполнены, на вопросы отвечает используя интернет
15-19	Не все требования программы выполнены, но ориентируется в коде.
10-14	Не все требования программы выполнены, в коде ориентируется плохо.
0-9	Программа реализована частично

### 3) Требования к защите проекта:

- 1) Наличие пояснительной записки
  - a) Задача
  - b) Ссылка на соревнование
  - c) Описание структуры данных
  - d) Описание использованных методов
  - e) Оценка полученных результатов
  - f) Ссылка на программный код
  - g) Таблица результатов соревнования
  - h) Ссылка на презентацию
- 2) Презентация итогов
  - a) Полнота
  - b) Ясность
  - c) Оформление
- 3) Защита проекта
  - a) Качество речи
  - b) Понимание задачи
  - c) Понимание хода решения
  - d) Ответы на вопросы

### Примерные соревнования:

- 1) [Kaggle.com](https://kaggle.com)
- 2) <https://yandex.ru/cup/>
- 3) <https://nti-contest.ru/>
- 4) <https://sibur.ai-community.com/>

1.

Оценка	Критерий оценки проекта
1-5	Уровень соревнования: начинающие, учебное, хакатон, профессиональное с ценным призом
0-30	30 баллов начисляется, если результаты проекта вошли в топ 30% от участников в соревновании

### БАЛЛЬНАЯ СТРУКТУРА ОЦЕНКИ

Название этапа	баллы
Работа с MOOK	25
Коллоквиум	25
Разработка проекта на конкурс ML	35
Защита проекта	15
<b>ЗАЧЕТ</b>	<b>&gt; 70</b>

### Показатели и критерии оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Уровень сформированности компетенций			
«Минимальный уровень не достигнут» (менее 55 баллов)	«Минимальный уровень» (56-70 баллов)	«Средний уровень» (71-85 баллов)	«Высокий уровень» (86-100 баллов)
<u>Компетенции не сформированы.</u>  Знания отсутствуют, умения и навыки не сформированы.	<u>Компетенции сформированы.</u>  Сформированы базовые структуры знаний. Умения фрагментарны и носят репродуктивный характер. Демонстрируется низкий уровень самостоятельности практического навыка.	<u>Компетенции сформированы.</u>  Знания обширные, системные. Умения носят репродуктивный характер, применяются к решению типовых заданий. Демонстрируется достаточный уровень самостоятельности устойчивого практического навыка.	<u>Компетенции сформированы.</u>  Знания твердые, аргументированные, всесторонние. Умения успешно применяются к решению как типовых, так и нестандартных творческих заданий. Демонстрируется высокий уровень самостоятельности, высокая адаптивность практического навыка
Описание критериев оценивания			
Обучающийся демонстрирует: - существенные пробелы в знаниях учебного материала; - допускаются принципиальные ошибки при ответе на основные вопросы, отсутствует знание и понимание основных понятий и категорий; - непонимание сущности дополнительных вопросов в рамках заданий;	Обучающийся демонстрирует: - знания теоретического материала; - неполные ответы на основные вопросы, ошибки в ответе, недостаточное понимание сущности излагаемых вопросов; - неуверенные и неточные ответы на дополнительные вопросы; - недостаточное владение литературой, рекомендованной	Обучающийся демонстрирует: - знание и понимание основных вопросов контролируемого объема программного материала; - твердые знания теоретического материала. - способность устанавливать и объяснять связь практики и теории, выявлять противоречия, проблемы и тенденции развития;	Обучающийся демонстрирует: - глубокие, всесторонние и аргументированные знания программного материала; - полное понимание сущности и взаимосвязи рассматриваемых процессов и явлений, точное знание основных понятий в рамках обсуждаемых заданий; - способность устанавливать и

<p>- отсутствие умения выполнять практические задания, предусмотренные программой дисциплины;</p> <p>- отсутствие готовности (способности) к дискуссии и низкую степень контактности.</p>	<p>программой дисциплины;</p> <p>- умение без грубых ошибок решать практические задания, которые следует выполнить.</p>	<p>- правильные и конкретные, без грубых ошибок, ответы на поставленные вопросы;</p> <p>- умение решать практические задания, которые следует выполнить;</p> <p>- владение основной литературой, рекомендованной программой дисциплины;</p> <p>- наличие собственной обоснованной позиции по обсуждаемым вопросам. Возможны незначительные оговорки и неточности в раскрытии отдельных положений вопросов, присутствует неуверенность в ответах.</p>	<p>объяснять связь практики и теории;</p> <p>- логически последовательные, содержательные, конкретные и исчерпывающие ответы на все задания, а также дополнительные вопросы экзаменатора;</p> <p>- умение решать практические задания;</p> <p>- свободное использование в ответах на вопросы материалов рекомендованной основной и дополнительной литературы.</p>
<b>Оценка «неудовлетворительно» / не зачтено</b>	<b>Оценка «удовлетворительно» / «зачтено»</b>	<b>Оценка «хорошо» / «зачтено»</b>	<b>Оценка «отлично» / «зачтено»</b>

## 9. Учебно-методическое и информационное обеспечение дисциплины

### а) основная литература:

- 1) "Поколение Python" (BEEGEEK) - слишком простой <https://stepik.org/course/58852/syllabus>
- 2) "Программирование на Python" (Институт БИОИНФОРМАТИКИ ИТМО) - простой <https://stepik.org/course/67/syllabus>
- 3) "Python: основы и применение" (Институт БИОИНФОРМАТИКИ ИТМО) - продвинутый <https://stepik.org/lesson/24459/step/1?unit=6764>
- 4) "Adaptive Python" (JetBrains, англ.) - сложность задач подбирается под ваш уровень <https://stepik.org/course/568>
- 5) "Основы программирования на Python" (ВШЭ) - норм <https://www.coursera.org/learn/python-osnovy-programm..>
- 6) "Поружение в Python" (МФТИ) - норм <https://www.coursera.org/learn/diving-in-python#sylla..>
- 7) "Основы Python и анализа данных" (Yandex практикум, частично бесплатный) - норм <https://praktikum.yandex.ru/profile/data-analyst/>
- 8) Сбербанк. Онлайн курс по машинному обучению <https://ml.ai-academy.ru/>

### б) дополнительная литература:

- 1) Англоязычный портал по машинному обучению <https://www.datacamp.com/about>

2) Open Machine Learning Course <https://mlcourse.ai/>

3)

**в) электронные библиотечные системы, с которыми у СОГУ имеется действующий договор, современные профессиональные базы, информационные справочные системы:**

– eLIBRARY.RU [Электронный ресурс]: научная электронная библиотека. – URL: <http://www.elibrary.ru>.

– База данных «ЭБС elibrary»: <http://elibrary.ru>

– Издательство «Юрайт» [Электронный ресурс]: электронно-библиотечная система. – URL: <http://biblio-online.ru>.

– Университетская библиотека online [Электронный ресурс]: электронно-библиотечная система. – URL: <http://www.biblioclub.ru>.

– Агрегатор онлайн курсов <https://www.coursera.org/>

– Портал соревнований по машинному обучению [Kaggle Dataset](#).

## **10. Материально-техническое обеспечение дисциплины**

Занятия по дисциплине проводятся в аудиториях, обеспеченных компьютерами, имеющими доступ к сети Интернет, интерактивными досками и мультимедийным оборудованием.

*Лицензионное программное обеспечение:*

1. Windows 10 Pro for Workstations, (№ 4100072800 Microsoft Products (MPSA) от 04.2016г);
2. Office Standard 2016 (№ 4100072800 Microsoft Products (MPSA) от 04.2016г);
3. Система поиска текстовых заимствований «Антиплагиат ВУЗ»;

*Перечень ПО в свободном доступе:*

1. Kaspersky Free;
2. WinRar;
3. Google Chrome;
4. Yandex Browser;
5. OperaBrowser;
6. VisualStudioCode;
7. Anaconda3;
8. PyCharm-community;
9. Python 3.8.5;
10. Sublime text 3;
11. Collab
12. Wing 101

## **11. Лист обновления/актуализации**

1. Рабочая программа

пересмотрена и актуализирована на заседании кафедры алгебры и геометрии протокол № 7 от 24.03.2020г.;

одобрена на заседании совета факультета математики и информационных технологий, протокол № 5 от 27.03.2020 г.