

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

УТВЕРЖДАЮ

Проректор по учебной работе,
качеству образования – первый
проректор

Хагуров Т.А.

подпись

«30» мая 2025

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.01 «Основы программирования на Python»

Направление подготовки 09.03.03 Прикладная информатика

Направленность (профиль) Прикладная информатика в экономике

Форма обучения очная

Квалификация бакалавр

Краснодар 2025

Рабочая программа дисциплины «Основы программирования на Python» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 09.03.03 Прикладная информатика.

Программу составил(и):

Харченко Анна Владимировна, доцент, канд. пед наук

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



подпись

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

Заведующий кафедрой (разработчика)

В. В. Подколзин



подпись

Рабочая программа обсуждена на заседании кафедры анализа данных и искусственного интеллекта протокол №13 от «20» мая 2025 г.

Заведующий кафедрой (выпускающей)

А. В. Коваленко

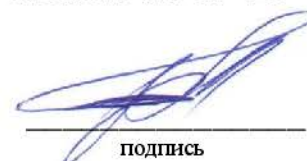


подпись

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

Председатель УМК факультета

А. В. Коваленко



подпись

Рецензенты:

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

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

1 Цели и задачи изучения дисциплины (модуля)

1.1 Цель освоения дисциплины

Целью дисциплины является формирование у студентов фундаментальных знаний и практических навыков разработки программ на языке Python, включая понимание его синтаксиса, основных структур данных, функций, работы с файлами, а также применение специализированных библиотек для обработки данных и визуализации.

1.2 Задачи дисциплины

- изучить особенности интерпретируемых языков на примере Python;
- освоить базовый синтаксис и конструкции языка;
- научиться работать с основными структурами данных;
- развить навыки создания функций и модулей;
- освоить методы работы с файлами и регулярными выражениями;
- изучить основы NumPy для вычислений и Matplotlib для визуализации.

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Основы программирования на Python» относится к «Часть, формируемая участниками образовательных отношений» Блока 1 «Дисциплины (модули)» учебного плана.

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

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

ПК-1 Способен решать актуальные и значимые задачи прикладной информатики

- Знать** Основные принципы алгоритмизации и проектирования программных решений
Методологию разработки и реализации вычислительных алгоритмов
Теоретические основы обработки структурированных данных
Современные подходы к анализу и визуализации информации
Принципы тестирования и верификации программного обеспечения
- Уметь** Формализовывать предметную область и ставить вычислительные задачи
Проектировать эффективные алгоритмы для решения прикладных задач
Реализовывать программные решения с использованием современных инструментов
Обрабатывать и анализировать данные различной природы
Визуализировать результаты вычислений для интерпретации
- Владеть** Навыками разработки алгоритмических решений
Методами обработки и анализа структурированных данных
Технологиями программной реализации вычислительных методов
Навыками работы со специализированными инструментальными средствами
Способностью адаптировать известные решения к новым задачам

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 2 зач. ед. (72 часа), их распределение по видам работ представлено в таблице

Вид учебной работы	Всего часов	Семестры (часы)
---------------------------	--------------------	------------------------

		4					
Контактная работа, в том числе:	36,2	36,2					
Аудиторные занятия (всего):	34	34					
Занятия лекционного типа	16	16					
Лабораторные занятия	18	18					
Занятия семинарского типа (семинары, практические занятия)							
Иная контактная работа:	2,2	2,2					
Контроль самостоятельной работы (КСР)	2	2					
Промежуточная аттестация (ИКР)	0,2	0,2					
Самостоятельная работа, в том числе:	35,8	35,8					
<i>Проработка учебного (теоретического) материала</i>	12	12					
<i>Выполнение индивидуальных заданий (подготовка сообщений, презентаций)</i>	20	20					
Подготовка к текущему контролю	3,8	3,8					
Контроль:							
Подготовка к экзамену							
Общая трудоемкость	час.	72	72				
	в том числе контактная работа	36,2	36,2				
	зач. ед	2	2				

2.2 Структура дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

Разделы (темы) дисциплины, изучаемые в 4 семестре

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа СРС
			Л	ПЗ	ЛР	
1	2	3	4	5	6	7
1.	Интерпретируемые языки программирования и их особенности	6	2		2	2
2.	Кортежи и списки	8	2		2	4
3.	Множества	10	2		2	6
4.	Словари	8	2		2	4
5.	Функции и исключения	10	2		2	6
6.	Файлы	12	2		4	6
7.	Регулярные выражения	8	2		2	4
8.	Пакеты numpy, matplotlib	7,8	2		2	3,8
ИТОГО по разделам дисциплины		69,8	16		18	35,8
Контроль самостоятельной работы (КСР)		2				
Промежуточная аттестация (ИКР)		0,2				
Подготовка к текущему контролю						
Общая трудоемкость по дисциплине		72				

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

2.3 Содержание разделов (тем) дисциплины

2.3.1 Занятия лекционного типа

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
1.	1. Интерпретируемые языки программирования и их особенности	- Понятие интерпретируемых и компилируемых языков. Виртуальные машины и байт-код - Особенности Python: динамическая типизация, управление памятью - Базовый синтаксис: переменные, операторы, основные типы данных - Управляющие конструкции: условия, циклы - Интерактивные среды: IPython, Jupyter Notebook, Google Colab	
2.	2. Кортежи и списки	- Создание и основные операции со списками (list) - Методы списков: добавление, удаление, сортировка элементов - Срезы и итерация по спискам - Особенности кортежей (tuple), отличия от списков - Применение кортежей для хранения неизменяемых данных	
3.	3. Множества	- Создание множеств (set) - Операции над множествами: объединение, пересечение, разность - Методы работы с множествами - Особенности хэшируемых объектов - Применение множеств для удаления дубликатов	
4.	4. Словари	- Структура словарей (dict) - Основные операции: добавление, изменение, удаление элементов - Методы словарей: keys(), values(), items() - Особенности хэш-таблиц - Применение словарей для хранения структурированных данных	
5.	5. Функции и исключения	- Создание и вызов функций - Параметры и возвращаемые значения - Области видимости переменных - Обработка исключений: try-except-finally - Пользовательские исключения - Рекурсивные функции	
6.	6. Файлы	- Открытие и закрытие файлов - Режимы работы с файлами (чтение, запись, добавление) - Текстовые и бинарные файлы - Работа с CSV-файлами - Модуль os для работы с файловой системой	
7.	7. Регулярные выражения	- Синтаксис регулярных выражений - Модуль re: поиск, замена, разбиение строк - Специальные символы и квантификаторы - Группы и обратные ссылки - Применение для обработки текстовых данных	
8.	8. Пакеты numpy, matplotlib	- Создание и работа с массивами numpy - Базовые операции линейной алгебры - Визуализация данных: графики, диаграммы, гистограммы - Настройка отображения графиков - Сохранение результатов визуализации	

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

2.3.2 Занятия семинарского типа

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

2.3.3 Лабораторные занятия

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
1.	Интерпретируемые языки программирования и их особенности	Основы работы в Python: синтаксис и управляющие конструкции	РЗ
2.	Кортежи и списки	Работа с последовательностями: списки и кортежи	РЗ
3.	Множества	Операции над множествами	РЗ
4.	Словари	Работа со словарями	РЗ
5.	Функции и исключения	Лямбда-функции. Генераторы. Итераторы	РЗ
6.	Файлы	Текстовые файлы. Файлы CSV.	РЗ
7.	Регулярные выражения	Обработка текста с использованием регулярных выражений	РЗ
8.	Пакеты numpy, matplotlib	Основы работы с NumPy: массивы и операции	РЗ
9.	8Пакеты numpy, matplotlib	Построение графиков и диаграмм с Matplotlib	РЗ

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

2.3.4 Примерная тематика курсовых работ (проектов)

Не предусмотрены

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

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

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

Для лиц с нарушениями зрения:

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

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

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

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. Образовательные технологии

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

При обучении используются следующие образовательные технологии:

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

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

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

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

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

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

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

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

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

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

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

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

Основные виды интерактивных образовательных технологий включают в себя:

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

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

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

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

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

Семестр	Вид занятия	Используемые интерактивные образовательные технологии	количество интерактивных часов
3	ЛР	Практические занятия в режимах взаимодействия «преподаватель – студент» и «студент – студент»	4
Итого			4

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

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

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

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

Для лиц с нарушениями зрения:

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

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

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

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

4. Оценочные и методические материалы

4.1 Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

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

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

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

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

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

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

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

Для лиц с нарушениями зрения:

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

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

Структура оценочных средств для текущей и промежуточной аттестации

№ п/п	Контролируемые разделы (темы) дисциплины*	Код контролируемой компетенции (или ее части)	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
1	Интерпретируемые языки программирования и их особенности	ПК-1	Типовые контрольные задания 1-3 Типовые тестовые задания 1-2	Задание для самостоятельной работы и промежуточной аттестации 1-2
2	Кортежи и списки	ПК-1	Типовые контрольные задания 4-6 Типовые тестовые задания 3-4	Задание для самостоятельной работы и промежуточной аттестации 3-4

3	Множества	ПК-1	Типовые контрольные задания 7-9 Типовые тестовые задания 5-6	Задание для самостоятельной работы и промежуточной аттестации 5-6
4	Словари	ПК-1	Типовые контрольные задания 10-12 Типовые тестовые задания 7-8	Задание для самостоятельной работы и промежуточной аттестации 7-8
5	Функции и исключения	ПК-1	Типовые контрольные задания 13-15 Типовые тестовые задания 9-10	Задание для самостоятельной работы и промежуточной аттестации 9-10
6	Файлы	ПК-1	Типовые контрольные задания 16-18 Типовые тестовые задания 11-12	Задание для самостоятельной работы и промежуточной аттестации 11-12
7	Регулярные выражения	ПК-1	Типовые контрольные задания 19-21 Типовые тестовые задания 13-14	Задание для самостоятельной работы и промежуточной аттестации 13-14
8	Пакеты numpy, matplotlib	ПК-1	Типовые контрольные задания 21-23 Типовые тестовые задания 15-16	Задание для самостоятельной работы и промежуточной аттестации 15-16

Показатели, критерии и шкала оценки сформированных компетенций

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

ПК-1 Способен решать актуальные и значимые задачи прикладной информатики

- Знать** Основные принципы и методы алгоритмизации и программирования
Синтаксис и базовые конструкции языка Python
Особенности работы с основными структурами данных (списки, словари, множества, кортежи)
Основы работы с файлами и внешними источниками данных
Принципы использования библиотек для анализа данных (NumPy, Pandas) и визуализации (Matplotlib)
- Уметь** Разрабатывать алгоритмы для решения типовых задач прикладной информатики
Реализовывать программы на Python с использованием различных структур данных
Обрабатывать и анализировать данные из различных источников

	Применять специализированные библиотеки Python для решения прикладных задач
	Тестировать и отлаживать программный код
Владеть	Навыками программирования на Python для решения практических задач
	Методами обработки и анализа данных
	Технологиями визуализации результатов вычислений
	Навыками работы с инструментами разработки и отладки программ
	Способностью адаптировать существующие решения под новые задачи

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

Типовые тестовые задания

1. Какое утверждение верно для интерпретируемых языков (например, Python)?
 - а) Код компилируется в машинный язык перед выполнением
 - б) Код выполняется построчно специальной программой-интерпретатором
 - в) Они работают быстрее компилируемых языков
 - г) Не поддерживают динамическую типизацию
2. Какой инструмент НЕ является средой разработки для Python?
 - а) PyCharm
 - б) Jupyter Notebook
 - в) Microsoft Word
 - г) Visual Studio Code
3. Что из этого можно сделать со списком, но нельзя с кортежем?
 - а) Хранить элементы разных типов
 - б) Изменить элемент по индексу (например, `list[0] = 5`)
 - в) Перебрать элементы в цикле
 - г) Получить длину через `len()`
4. Как создать пустой кортеж?
 - а) `tuple = []`
 - б) `tuple = ()`
 - в) `tuple = tuple()`
 - г) Кортеж нельзя создать пустым
5. Какой код удалит дубликаты из списка `[1, 2, 2, 3]`?
 - а) `list.remove_duplicates()`
 - б) `list(set([1, 2, 2, 3]))`
 - в) `list.unique()`
 - г) `set(list.clear())`
6. Что выведет `{1, 2, 3} | {3, 4, 5}`?
 - а) `{3}`
 - б) `{1, 2, 3, 4, 5}`
 - в) `{1, 2, 4, 5}`
 - г) Ошибку, так как оператор `|` не поддерживается
7. Как получить список всех ключей словаря `dict = {"a": 1, "b": 2}`?
 - а) `dict.keys()`
 - б) `list(dict.keys())`
 - в) `dict.items()`
 - г) `dict.values()`
8. Что делает метод `dict.get("key", "default")`?
 - а) Удаляет ключ "key", если он есть
 - б) Возвращает значение по ключу "key" или "default", если ключа нет
 - в) Добавляет ключ "key" со значением "default"
 - г) Вызывает ошибку, если ключа нет

9. Какое исключение возникнет при делении на ноль?
- а) ValueError
 - б) ZeroDivisionError
 - в) TypeError
 - г) IndexError
10. Что выведет код?
- ```
def func(x, y=2):
 return x * y
print(func(3))
```
- а) 6
  - б) Ошибку, так как у не указан
  - в) 32
  - г) None
11. Какой режим откроет файл для записи с очисткой содержимого?
- а) "r"
  - б) "w"
  - в) "a"
  - г) "x"
12. Как прочитать все строки файла в список?
- а) file.read()
  - б) file.readlines()
  - в) file.readline()
  - г) file.read().splitlines()
13. Какое выражение найдёт все цифры в строке?
- а) re.findall("\a", text)
  - б) re.findall("\d", text)
  - в) re.findall("\w", text)
  - г) re.findall("\s", text)
14. Что заменит все пробелы на дефисы?
- а) re.sub(\_\_\_\_, "-", "Hello World")
  - б) " "
  - в) "\s"
  - г) "\w"
  - д) "."
15. Как создать массив NumPy из списка [1, 2, 3]?
- а) np.array({1, 2, 3})
  - б) np.array([1, 2, 3])
  - в) np.create\_array(1, 2, 3)
  - г) np.list\_to\_array(1, 2, 3)
16. Как построить график линии в Matplotlib?
- а) plt.bar(x, y)
  - б) plt.plot(x, y)
  - в) plt.pie(x)
  - г) plt.scatter(x, y)

### **Типовые контрольные задания**

1. Напишите программу, которая демонстрирует разницу во времени выполнения интерпретируемого кода Python и скомпилированного кода на C++ для одной и той же задачи (например, вычисление суммы чисел от 1 до N).

2. Создайте скрипт, который использует Jupyter Notebook для интерактивного ввода данных пользователем и визуализации результатов простых вычислений.
3. Напишите функцию, которая принимает список чисел и возвращает новый список, где каждый элемент умножен на 2, но исходный список остается неизменным.
4. Реализуйте программу, которая преобразует кортеж строк в кортеж чисел (если это возможно) и обрабатывает ошибки преобразования.
5. Создайте функцию, которая принимает два множества и возвращает их симметрическую разность, не используя встроенные операторы.
6. Напишите программу для проверки, являются ли два слова анаграммами, используя множества.
7. Реализуйте простую базу данных на словарях, позволяющую добавлять, удалять и искать записи по ключу.
8. Напишите функцию, которая инвертирует словарь (ключи становятся значениями и наоборот), обрабатывая случай повторяющихся значений.
9. Создайте декоратор, который измеряет и выводит время выполнения функции.
10. Напишите функцию, которая рекурсивно вычисляет факториал числа с обработкой исключений для отрицательных значений.
11. Реализуйте программу для объединения содержимого нескольких текстовых файлов в один с нумерацией строк.
12. Создайте скрипт, который анализирует CSV-файл, вычисляет статистику по числовым столбцам и сохраняет результат в новый файл.
13. Напишите программу для валидации email-адресов с использованием регулярных выражений.
14. Создайте скрипт, который ищет в тексте все даты в формате DD.MM.YYYY и заменяет их на формат YYYY-MM-DD.
15. Реализуйте программу, которая генерирует случайную матрицу с помощью NumPy и вычисляет ее определитель.
16. Создайте скрипт, который строит три типа графиков (линейный, столбчатый и круговую диаграмму) для одного набора данных с помощью Matplotlib.

### **Вопросы для подготовки к зачету к зачету**

1. В чем основные отличия интерпретируемых языков программирования от компилируемых? Приведите примеры.
2. Какие преимущества и недостатки имеет интерпретация кода по сравнению с компиляцией?
3. Опишите процесс выполнения Python-программы интерпретатором. Какие этапы он включает?
4. В чем принципиальная разница между списками и кортежами в Python? Когда следует использовать каждый из них?
5. Какие основные методы работы со списками вы знаете? Приведите примеры их использования.
6. Как работают срезы (slice) в Python? Продемонстрируйте на примере списка.
7. Каковы основные свойства множеств в Python? Чем они отличаются от списков?
8. Какие операции можно выполнять над множествами? Приведите примеры практического применения.
9. Как работает хеширование в множествах? Какие типы данных можно использовать как элементы множества?
10. Чем словари отличаются от других структур данных в Python? Каковы их основные характеристики?



11. Опишите основные методы работы со словарями. В каких задачах они наиболее полезны?
12. Как работает механизм хеш-таблиц в реализации словарей Python?
13. Какие виды параметров функций существуют в Python? Чем они отличаются?
14. Как работает механизм обработки исключений? Приведите примеры встроенных исключений.
15. Что такое область видимости переменных? Как работают локальные и глобальные переменные?
16. Какие основные режимы работы с файлами существуют в Python? В чем их отличия?
17. Как организовать чтение и запись данных в файлы разных форматов (текстовые, CSV)?
18. В чем преимущества использования менеджера контекста (with) при работе с файлами?
19. Каковы основные элементы синтаксиса регулярных выражений? Приведите примеры.
20. Как выполняется поиск и замена с использованием регулярных выражений в Python?
21. Каковы основные возможности библиотек NumPy и Matplotlib? В каких задачах они применяются?
22. Как создаются и обрабатываются массивы в NumPy? Какие операции с ними можно выполнять?
23. Какие типы визуализации данных поддерживает Matplotlib? Приведите примеры построения графиков.

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

##### **Методические рекомендации, определяющие процедуры оценивания тестов:**

Тест проводится онлайн в системе Moodle или Google Docs и ограничен по времени. На сдачу теста дается две попытки. Тест считается успешно пройденным если студент правильно ответил на 70% вопросов.

##### **Методические рекомендации, определяющие процедуры оценивания выполнения контрольных заданий:**

Задание считается выполненным при выполнении следующих условий:

- предоставлен исходный код на Python в среде PyCharm, Google Collab
- продемонстрирована работоспособность программы
- студент понимает исходный код и отвечает на вопросы по его организации.

##### **Методические рекомендации к сдаче зачета**

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

Текущий контроль успеваемости студентов проводится в целях совершенствования и непрерывного контроля качества образовательного процесса, проверки усвоения учебного материала, активизации самостоятельной работы студентов, стимулирования их учебной

работы, обеспечения эффективности образовательного процесса, предупреждения рисков отчисления студентов.

Текущий контроль знаний студентов осуществляется постоянно в течение всего семестра.

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

Студенты очной формы обучения обязаны сдать зачет до начала экзаменационной сессии.

По результатам сдачи зачета выставляется оценка «зачтено»/ «не зачтено».

Выставление зачетов для студентов очной формы обучения проводятся в период до экзаменационной сессии.

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

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

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

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

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

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

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

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

В качестве систем программирования для решения задач и изучения методов и алгоритмов, приведенных в лекциях, рекомендуется использовать на практических

занятиях и при самостоятельной работе стандартную реализацию языка Python в связке со средой разработки PyCharm, Google Collab.

| Оценка                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                            |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Не зачтено                                                                                                                                                                                                                                            | Зачтено                                                                                                                                                                                                                                                                                    |
| <ul style="list-style-type: none"><li>• если студент правильно решил менее 70 % задач и/или не имеет представление как решать остальные задачи</li><li>не знает значительной части теоретического материала, допускает существенные ошибки.</li></ul> | <ul style="list-style-type: none"><li>• если студент правильно решил 70 % задач, имеет представление как решать остальные задачи</li><li>• на теоретические вопросы дан развернутый ответ. Материал изложен в целом последовательно. Имеются логичные и аргументированные выводы</li></ul> |

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

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

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

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

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

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,

– в форме электронного документа.

Для лиц с нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

## **5. Перечень учебной литературы, информационных ресурсов и технологий**

### **5.1 Основная литература:**

1. Чернышев, С. А. Основы программирования на Python : учебное пособие для вузов / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 349 с. — (Высшее образование). — ISBN 978-5-534-17139-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/544190> (дата обращения: 08.06.2025).

2. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для среднего профессионального образования / Д. Ю. Федоров. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2021. — 210 с. — (Профессиональное образование). — ISBN 978-5-534-12829-1. — Текст : электронный // Образовательная



платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/487079> (дата обращения: 08.06.2025).

3. Гниденко, И. Г. Технология разработки программного обеспечения : учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 248 с. — (Профессиональное образование). — ISBN 978-5-534-18131-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/539215> (дата обращения: 08.06.2025).

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

### **5.2 Дополнительная литература:**

1. Сузи, Р. А. Язык программирования Python : учебное пособие : [16+] / Р. А. Сузи. — 2-е изд., испр. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ) : Бинوم. Лаборатория знаний, 2007. — 327 с. — (Основы информационных технологий). — Режим доступа: по подписке. — URL: <https://biblioclub.ru/index.php?page=book&id=233288> (дата обращения: 08.06.2025). — ISBN 978-5-9556-0109-0. — Текст : электронный.

2. Шелудько, В. М. Язык программирования высокого уровня Python: функции, структуры данных, дополнительные модули : [16+] / В. М. Шелудько. — Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2017. — 108 с. : ил. — Режим доступа: по подписке. — URL: <https://biblioclub.ru/index.php?page=book&id=500060> (дата обращения: 08.06.2025). — Библиогр. в кн. — ISBN 978-5-9275-2648-2. — Текст : электронный.

3. Программные системы статистического анализа: обнаружение закономерностей в данных с использованием системы R и языка Python : [16+] / В. М. Волкова, М. А. Семенова, Е. С. Четвертакова, С. С. Вожов. — Новосибирск : Новосибирский государственный технический университет, 2017. — 74 с. : ил., табл. — Режим доступа: по подписке. — URL: <https://biblioclub.ru/index.php?page=book&id=576496> (дата обращения: 08.06.2025). — Библиогр.: с. 48. — ISBN 978-5-7782-3183-2. — Текст : электронный.

### **5.3. Периодические издания:**

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

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

*Электронно-библиотечные системы (ЭБС):*

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

*Профессиональные базы данных*

1. Scopus <http://www.scopus.com/>
2. ScienceDirect <https://www.sciencedirect.com/>
3. Журналы издательства Wiley <https://onlinelibrary.wiley.com/>
4. Научная электронная библиотека (НЭБ) <http://www.elibrary.ru/>
5. Полнотекстовые архивы ведущих западных научных журналов на Российской платформе научных журналов НЭИКОН <http://archive.neicon.ru>
6. Национальная электронная библиотека (доступ к Электронной библиотеке диссертаций Российской государственной библиотеки (РГБ) <https://rusneb.ru/>
7. Президентская библиотека им. Б.Н. Ельцина <https://www.prilib.ru/>



8. База данных CSD Кембриджского центра кристаллографических данных (CCDC) <https://www.ccdc.cam.ac.uk/structures/>
9. Springer Journals: <https://link.springer.com/>
10. Springer Journals Archive: <https://link.springer.com/>
11. Nature Journals: <https://www.nature.com/>
12. Springer Nature **Protocols and Methods**:  
<https://experiments.springernature.com/sources/springer-protocols>
13. Springer Materials: <http://materials.springer.com/>
14. Nano Database: <https://nano.nature.com/>
15. Springer eBooks (i.e. 2020 eBook collections): <https://link.springer.com/>
16. "Лекториум ТВ" <http://www.lektorium.tv/>
17. Университетская информационная система РОССИЯ <http://uisrussia.msu.ru>

#### *Информационные справочные системы*

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

#### *Ресурсы свободного доступа*

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

#### *Собственные электронные образовательные и информационные ресурсы КубГУ*

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

## **5.5 Перечень информационно-коммуникационных технологий**

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

## **5.6 Перечень лицензионного и свободно распространяемого программного обеспечения**

OpenOffice  
 Компилятор C++  
 Oracle VirtualBox 6  
 VMware Workstation 16  
 Putty 0.76 или Kitty 0.76  
 FileZilla 3.57.0  
 WinSCP 5.19  
 Advanced port scanner 2.5  
 Python 3 (3.7 И 3.9)  
 numpy 1.22.0  
 opencv 4.5.5  
 Keras 2.7.0  
 Tensor flow 2.7.0  
 matplotlib 3.5.1  
 PyCharm 2021  
 Cuda Toolkit 11.6  
 Фреймворк Django  
 Firefox, любая версия  
 Putty, любая версия  
 Visual Studio Code, версия 1.52+  
 Eclipse PHP Development Tools, версия 2020-06+  
 Плагин Remote System Explorer (RSE) для Eclipse PDT  
 JetBrains PHP Storm  
 GIT  
 Java Version 8 Update 311  
 Clojure 1.10.3.1029.ps1  
 SWI Prolog 8.4  
 IntelliJ Idea IDE 2021  
 Mozilla Firefox 96  
 Google Chrome 97  
 GitHub Desktop 2.9  
 PHP Storm 2021  
 FileZilla 3.57.0  
 Putty 0.76

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

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



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

## **7. Материально-техническое обеспечение по дисциплине (модулю)**

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

Примечание: Конкретизация аудиторий и их оснащение определяется ОПОП.