
УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«25» мая 2026 г.
Протокол № 2

**Рабочая программа дисциплины (модуля)
«Основы Python»**

Направление подготовки: 38.04.05 Бизнес-информатика

Направленность (профиль) подготовки: Продуктовый менеджмент

Квалификация (степень) выпускника: магистр

Форма обучения: очная

Срок освоения программы: 2 года

Год набора: 2026

**Москва
2026**

Содержание

1. Краткая характеристика дисциплины (модуля)	3
2. Перечень планируемых результатов обучения.....	5
3. Тематический план.....	6
4. Содержание дисциплины (модуля).....	6
5. Учебно-методическое обеспечение	7
6. Материально-техническое обеспечение	7
7. Методические и оценочные материалы	9

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Основы Python» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – магистратура по направлению 38.04.05 Бизнес-информатика, профиль Продуктовый менеджмент, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 990 от 12.08.2020 года.

Изучение дисциплины (модуля) «Основы Python» помогает студентам освоить один из самых популярных и универсальных языков программирования, широко используемых в науке, бизнесе и IT-индустрии. Освоение основ Python позволяет эффективно автоматизировать задачи, анализировать данные и создавать программные продукты, что значительно расширяет профессиональные возможности.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки магистратуры по направлению 38.04.05 Бизнес-информатика, профиль Продуктовый менеджмент и входит в обязательную часть Блока 1.

Дисциплина (модуль) изучается на 1 курсе в 1 семестре, проводится в формате онлайн-курса.

Цель изучения дисциплины (модуля): формирование фундаментальных знаний и навыков программирования на Python, позволяющих студентам решать базовые задачи с использованием коллекций, функций и объектно-ориентированного подхода, обеспечивая качественный и поддерживаемый код.

Задачи изучения дисциплины (модуля):

- освоить теоретические основы Python;
- приобрести практические навыки работы с данными и функциями;
- освоить обработку ошибок и объектно-ориентированное программирование;
- развить навыки реализации программ и организации кода;
- применять принципы алгоритмизации и качественной разработки.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

- основные принципы работы с переменными и их типами в Python;
- арифметические и логические операции, их синтаксис и применение;
- понятие функций, их назначение и синтаксис объявления;
- основные коллекции в Python: строки, списки, словари, множества и кортежи;
- принципы работы с файлами: чтение, запись, работа с текстовыми файлами;
- области видимости переменных в контексте функций и модулей;
- различия между процедурным и объектно-ориентированным подходами.

уметь:

- использовать встроенные методы и операции для работы со строками, списками, словарями, множествами и кортежами;
- писать и вызывать функции, включая функции с параметрами и возвратом значений;
- обрабатывать файлы: открывать, читать, записывать данные и управлять файлами через менеджеров контекста (with);
- выполнять базовую отладку кода и использовать рекомендации по написанию читаемого и поддерживаемого кода;
- создавать срезы для строк и списков, использовать итерации и циклы для работы с коллекциями;
- использовать основные конструкции Python для обработки ошибок (try-except);

- определять собственные классы, создавать экземпляры объектов, писать методы класса и использовать `__init__` для инициализации объектов.

владеть:

- навыками реализации программ для решения задач базового уровня сложности с использованием коллекций и функций;
- навыками организации кода в небольшие модули, используя функции для структурирования программы;
- навыками автоматизации обработки данных из файлов;
- навыками эффективного использования коллекции для хранения и обработки данных;
- навыками применения базовых принципов алгоритмизации при решении задач (например, работа с последовательностями, поиск, сортировка);
- навыками разработки решений, соответствующих требованиям к качеству кода (PER 8);
- навыками реализации программ в объектно-ориентированной парадигме.

2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

Компетенция	Содержание компетенции	Индикатор компетенции	Перечень планируемых результатов обучения по дисциплине (модулю)
ОПК-2.	Способен учитывать конкретные условия выполняемых задач и разрабатывать инновационные решения при управлении проектами и процессами в сфере информационно-коммуникационных технологий	ОПК-2.1.	Знает методы анализа внешней и внутренней среды проекта, а также современные подходы к управлению ИКТ-проектами и бизнес-процессами
		ОПК-2.2.	Умеет разрабатывать и внедрять инновационные решения с учётом специфики задач, ограничений и потребностей заинтересованных сторон
		ОПК-2.3.	Имеет практический опыт применения гибких и классических методологий управления проектами в условиях неопределённости и быстро меняющейся технологической среды
ПК-2.	Способен разрабатывать и внедрять цифровые продукты с учетом принципов устойчивого развития и цифровой инклюзивности	ПК-2.1	Знает принципы устойчивого развития и цифровой инклюзивности, а также их применение в процессе создания цифровых продуктов
		ПК-2.2	Умеет проектировать цифровые решения, учитывающие экологические, социальные и этические аспекты, включая доступность для различных категорий пользователей
		ПК-2.3	Имеет практический опыт внедрения цифровых продуктов с соблюдением стандартов инклюзивного дизайна и оценки их воздействия на устойчивое развитие

3. Тематический план

№ п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы			ТКУ (текущий контроль успеваемости)
		Очная форма			
		Аудиторная работа	Контроль	Самостоя тельная работа	
		Семинары			
1	Синтаксис и базовые конструкции	1		4	Контест
2	Циклы и управление потоком исполнения	1		4	Контест Домашние задания
3	Структуры данных	2		5	Контест Домашние задания
4	Функции и области видимости	2		5	Контест Домашние задания
5	Работа с файлами и обработка ошибок	2		5	Контест Домашние задания
6	Основы объектно-ориентированного программирования	2		5	Проект
	Зачет				
Итого:		10		28	
Объем дисциплины (модуля) (в ак. ч.)		38			
Объем дисциплины (модуля) (в зач. ед.)		1			

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Синтаксис и базовые конструкции	Основы синтаксиса и типы данных Условные конструкции
2	Циклы и управление потоком исполнения	Циклы: for и while
3	Структуры данных	Списки, кортежи и строки Словари и множества
4	Функции и области видимости	Функции и области видимости
5	Работа с файлами и обработка ошибок	Работа с файлами и обработка ошибок
6	Основы объектно-ориентированного программирования	Объектно-ориентированное программирование

5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

Основная литература:

1. Чернышев, С. А. Основы программирования на Python : учебник для вузов / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 349 с. — (Высшее образование). — ISBN 978-5-534-17139-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/567821>.

Дополнительная литература:

1. Федоров, Д. Ю. Программирование на python : учебное пособие для вузов / Д. Ю. Федоров. — 6-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 187 с. — (Высшее образование). — ISBN 978-5-534-19666-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/556864>.

6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

- столами и стульями;
- компьютерной техникой;
- механическими калькуляторами;
- специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Наименование портала (издания, курса, документа)	Ссылка
1	Катастрофы, стихийные бедствия, аварии, эпидемии. Солнечная и геомагнитная активность. /ежедневный обзор	http://www.disasters.chat.ru
2	Каталог по безопасности жизнедеятельности	http://www.eun.chat.ru
3	Научная электронная библиотека eLibrary.ru библиотека	https://elibrary.ru/defaultx.asp
4	База данных для IT-специалистов	https://habr.com
5	База данных ScienceDirect	https://www.sciencedirect.com
6	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
7	Федеральный портал «Российское образование»	https://www.edu.ru/
8	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
9	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
10	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
Операционные системы:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		
Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное
Системы управления базами данных:		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Средства антивирусной защиты:		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
Пакеты программных средств и библиотек:		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		

Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

7. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Основы Python» в рамках текущего контроля успеваемости используются такие виды учебной работы, как семинары, контесты, домашние задания, проект, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Семинар – это форма учебной деятельности, проводимая в учебном заведении под руководством преподавателя, где студенты активно участвуют в обсуждениях, практических заданиях и других формах взаимодействия.

Для успешной подготовки к семинару рекомендуется заранее ознакомиться с темой занятия и основными материалами, чтобы иметь возможность активно участвовать в обсуждении. Также полезно подготовить вопросы и идеи для обсуждения, что поможет глубже понять материал и продемонстрировать заинтересованность.

Контеcт – интерактивная платформа с заданиями разного уровня сложности и автоматической проверкой результатов.

Контеcт позволяет оперативно оценивать усвоение материала и выявлять пробелы в знаниях через тесты и практические задачи. Такой формат способствует регулярной самопроверке и повышает мотивацию к изучению дисциплины.

Домашнее задание – набор задач по темам недели.

При работе над домашними заданиями важно внимательно ознакомиться с требованиями и сроками выполнения. Рекомендуется разбивать задания на этапы, чтобы избежать перегрузки и лучше усвоить материал, использовать различные источники информации, включая учебники и онлайн-ресурсы, для более глубокого понимания темы.

Проект – исследовательская работа по дисциплине (модулю) и презентация результатов.

Для успешной подготовки к проекту рекомендуется: четко определить цели и задачи проекта; составить план работы, разбив проект на этапы с указанием сроков выполнения каждого из них; использовать разнообразные источники информации и инструменты для исследования темы; регулярно проверять прогресс и вносить коррективы в план, если это необходимо.

Самостоятельная работа – работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

Система оценивания результатов обучения по дисциплине (модулю)

Критерии получения уровня и оценивания сформированности компетенций по дисциплине (модулю) «Основы Python».

Оценивание уровня учебных достижений обучающихся по дисциплине (модулю) осуществляется в виде текущего контроля успеваемости.

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме *зачета*, при этом проводится оценка компетенций, сформированных по дисциплине.

Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом:

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	Зачтено	Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину (модуль). Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
9	Отлично	Зачтено	
8	Отлично	Зачтено	
7	Хорошо	Зачтено	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает
6	Хорошо	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
5	Удовлетворительно	Зачтено	Студент обладает базовыми знаниями по дисциплине (модулю), но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
4	Удовлетворительно	Зачтено	
3	Не сдан	Не зачтено	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	Не зачтено	
1	Не сдан	Не зачтено	

Зачет по дисциплине (модулю) «Основы Python» можно получить одним из способов:

- Решить 4 конкурса на 7 и более баллов.
- Решить 4 конкурса на 6 и более баллов и 2 домашних задания на 7 и более баллов.
- Решить 4 конкурса на 5 и более баллов и все домашние задания на 7 и более баллов
- Решить проект на 7 и более баллов.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные задания для конкурса

Конкурс 1. Анализ последовательности чисел

Темы: синтаксис, условные конструкции, циклы, списки

Условие:

На вход подаётся строка из целых чисел, разделённых пробелами.

Необходимо:

1. Вывести количество положительных чисел.
2. Вывести сумму чётных чисел.
3. Вывести максимальное число.

Формат входных данных:

Одна строка с числами.

Формат выходных данных:

Три числа через пробел:

<количество_положительных> <сумма_четных> <максимум>

Пример:

Вход:

1 -2 3 4 -5 6

Выход:

4 8 6

Критерии автопроверки:

- корректная обработка отрицательных чисел;
- отсутствие ошибок при пустом вводе (вывести 0 0 0);
- решение использует цикл.

Контекст 2. Частотный словарь слов

Темы: строки, словари, циклы, функции

Условие:

На вход подаётся строка текста.

Необходимо:

1. Привести текст к нижнему регистру.
2. Удалить знаки препинания (.,!?:;).
3. Подсчитать частоту каждого слова.
4. Вывести слово, которое встречается чаще всего, и его количество.

Если таких слов несколько — вывести лексикографически минимальное.

Формат входных данных:

Одна строка.

Формат выходных данных:

<слово> <количество>

Пример:

Вход:

Python is great, and python is powerful!

Выход:

python 2

Критерии автопроверки:

- корректная нормализация регистра;
- правильная работа со словарём;
- корректная обработка равной частоты.

Контекст 3. Класс "Банковский счёт"

Темы: функции, обработка ошибок, ООП

Условие:

Реализуйте класс BankAccount, который:

- принимает начальный баланс;
- имеет методы deposit(amount) и withdraw(amount);
- при попытке снять больше средств, чем есть на счёте, выбрасывает исключение;
- имеет метод get_balance().

Затем:

- создайте объект;
- обработайте список операций (вводится построчно: deposit 100, withdraw 50);
- выведите итоговый баланс;
- при ошибке выведите ERROR.

Формат входных данных:

Первая строка — начальный баланс.

Далее список операций (до конца ввода).

Формат выходных данных:

Итоговый баланс или ERROR.

Примерные домашние задания**Домашнее задание 1: "Первые шаги в Python"**

Темы: Синтаксис, типы данных, условные конструкции

1. Напишите программу, которая запрашивает у пользователя его имя и возраст. Если $\text{возраст} \geq 18$ — выведите: "Добро пожаловать, {имя}!". Иначе — "Извините, вход только с 18 лет."
2. Создайте переменные:
a = 15, b = 4.
Выведите результат: сложения, вычитания, умножения, деления, целочисленного деления и остатка.
3. Напишите программу, определяющую, является ли число чётным. Пользователь вводит число — программа выводит True или False.
4. Напишите калькулятор:
 - Запросите два числа и операцию (+, -, *, /)
 - Выведите результат.
 - Если операция не поддерживается — выведите "Неизвестная операция".
 - При делении на ноль — "Ошибка: деление на ноль".
5. Дополнительно: определите, является ли введённое число положительным, отрицательным или нулём. Выведите соответствующее сообщение.

Домашнее задание 2: "Работа с данными и циклами"

Темы: Циклы, списки, строки, множества, словари

1. Напишите цикл while, который запрашивает числа у пользователя до тех пор, пока не будет введено 0. После завершения — выведите сумму всех введённых чисел.
2. Создайте список из 10 случайных чисел (можно вручную). Напишите программу, которая:
 - Выводит все чётные числа
 - Находит максимальное и минимальное
 - Выводит сумму и среднее арифметическое
3. Дан список строк: ["apple", "banana", "apple", "cherry", "banana", "date"]. Удалите дубликаты и выведите уникальные значения с использованием множества.
4. Создайте словарь с информацией о трёх студентах: ключ — имя, значение — список оценок. Напишите цикл, который вычисляет средний балл каждого и выводит:
Имя: средний балл.
5. Напишите программу, которая подсчитывает количество каждой буквы в строке (например, text = "hello world"). Используйте словарь: ключ — буква, значение — количество. Пробелы игнорируйте.

Домашнее задание 3: "Функции, файлы и ООП"

Темы: Функции, обработка ошибок, работа с файлами, ООП

1. Напишите функцию is_palindrome(s), которая проверяет, является ли строка палиндромом (читается одинаково с начала и с конца). Учитывайте регистр и пробелы: приведите строку к нижнему регистру и удалите пробелы.
2. Напишите функцию read_file_lines(filename), которая:
 - Открывает файл и читает строки
 - Выводит их с номером (1, 2, 3...)
 - Обработывает FileNotFoundError — выводит "Файл не найден".
3. Создайте функцию save_to_log(data, filename="log.txt"), которая:
 - Добавляет строку data в конец файла
 - Добавляет метку времени (можно просто строку "2025-04-05")

4. Создайте класс `Product` с атрибутами: `name`, `price`, `quantity`. Добавьте метод `get_total_cost()`, возвращающий стоимость (цена $\times$ количество).
5. Создайте класс `Store`, содержащий список продуктов. Реализуйте методы:
 - `add_product(product)` — добавляет продукт
 - `get_total_value()` — возвращает общую стоимость всех товаров
 - `show_products()` — выводит все товары в формате: "Яблоко — 50 руб (3 шт)"

Примерное описание задания и критерии оценивания к проекту

Описание задания:

Создайте консольное приложение — простой мессенджер, в котором пользователи могут регистрироваться, отправлять сообщения, просматривать ленту и получать статистику активности. Проект должен демонстрировать владение базовыми и средними возможностями Python: ООП, работа с файлами, обработка ошибок, структуры данных.

Функционал:

- Регистрация пользователя (имя, никнейм)
- Отправка сообщения (от кого, кому, текст, метка времени)
- Просмотр всех сообщений или переписки с конкретным пользователем
- Просмотр топ-5 самых активных отправителей
- Данные сохраняются в JSON при выходе и загружаются при запуске
- Обработка ошибок: например, отправка сообщения несуществующему пользователю

Критерии оценивания:

- Реализован базовый функционал (регистрация, отправка, просмотр)
- Использованы классы и методы (ООП)
- Данные корректно сохраняются и считываются из файла
- Обработаны исключения и некорректный ввод
- Код чистый, с комментариями и логичной структурой

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Какая структура данных является неизменяемой последовательностью?	Кортеж (tuple)	ОПК-2
2.	Как называется правило разрешения имен переменных в Python?	LEGB (Local, Enclosing, Global, Built-in)	ОПК-2
3.	Какой блок выполняется всегда при обработке исключений?	finally	ОПК-2
4.	Зачем используется контекстный менеджер (with)?	Гарантированное закрытие ресурса	ОПК-2

5.	В чем риск изменяемого объекта как аргумента по умолчанию?	Сохранение состояния между вызовами	ОПК-2
6.	Какая сложность поиска ключа в словаре (dict) в среднем случае?	$O(1)$	ОПК-2
7.	Как создать глубокую копию вложенного объекта?	<code>copy.deepcopy()</code>	ОПК-2
8.	Какой метод возвращает строковое представление объекта для отладки?	<code>repr</code>	ПК-2
9.	Для чего используется ключевое слово <code>yield</code> в функции?	Создание генератора	ПК-2
10.	Как избежать утечки памяти при чтении большого файла?	Построчная итерация	ОПК-2
11.	Чем отличается <code>list</code> от <code>tuple</code> помимо изменяемости?	Семантика использования (набор vs запись)	ПК-2
12.	Что происходит при ошибке в блоке <code>try</code> до <code>except</code> ?	Поиск подходящего обработчика исключений	ПК-2
13.	Как объявить приватный атрибут в классе Python?	Префикс <code>__</code> (два подчеркивания)	ПК-2
14.	Какая кодировка рекомендуется для текстовых файлов по умолчанию?	UTF-8	ОПК-2
15.	Что возвращает функция <code>range</code> в Python 3?	Объект-итератор (range object)	ОПК-2
16.	Как передать переменное число именованных аргументов?	<code>**kwargs</code>	ПК-2
17.	Почему не рекомендуется модифицировать список во время итерации?	Пропуск элементов или смещение индексов	ПК-2
18.	Что такое утиная типизация (Duck Typing)?	Важность поведения, а не типа объекта	ПК-2
19.	Как проверить принадлежность объекта к классу или его подклассу?	<code>isinstance()</code>	ОПК-2
20.	Какой метод вызывается при инициализации экземпляра класса?	<code>__init__</code>	ПК-2