
УТВЕРЖДЕНА

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

**Рабочая программа дисциплины (модуля)
«Язык программирования Python»**

Направление подготовки: 02.04.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Программная инженерия в
ритейле

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.....	3
3. Тематический план.....	7
4. Содержание дисциплины (модуля).....	7
5. Учебно-методическое обеспечение	9
6. Материально-техническое обеспечение	9
7. Методические и оценочные материалы	11

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Язык программирования Python» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – магистратура по направлению 02.04.01 Математика и компьютерные науки, профиль Программная инженерия в ретейле, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 810 от 23.08.2017 года.

Изучение дисциплины (модуля) «Язык программирования Python» позволяет сформировать у студентов системные знания и практические навыки программирования на языке Python. Дисциплина (модуль) охватывает основы языка, объектно-ориентированное программирование, функциональные возможности (декораторы, генераторы), работу с контекстными менеджерами, а также практики тестирования и эффективной обработки данных.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки магистратуры по направлению 02.04.01 Математика и компьютерные науки, профиль Программная инженерия в ретейле и входит в вариативную часть Блока 1, формируемую участниками образовательных отношений как дисциплина по выбору. Дисциплина (модуль) доступна к изучению на 1 курсе в 1 семестре.

Цель изучения дисциплины (модуля): сформировать у студентов прочную теоретическую и практическую базу для разработки, тестирования и сопровождения программных решений на языке Python, соответствующих современным стандартам и лучшим практикам.

Задачи изучения дисциплины (модуля):

- освоить синтаксис языка Java и базовые управляющие конструкции;
- изучить принципы объектно-ориентированного программирования и их реализацию в Java;
- научиться обрабатывать исключения и использовать механизмы try-catch-finally и try-with-resources;
- освоить работу с коллекциями и Stream API для эффективной обработки данных;
- развить навыки написания читаемого, структурированного и поддерживаемого кода с использованием современных возможностей Java.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

- основные принципы языка Python: модель объектов, динамическую типизацию, встроенные типы данных и коллекции;
- устройство функций в Python: параметры, области видимости, декораторы;
- принципы объектно-ориентированного программирования в Python: классы, наследование, полиморфизм, протоколы и абстрактные интерфейсы;
- принципы работы итераторов и генераторов;
- базовые принципы конкурентного программирования в Python: отличие конкурентности от параллелизма, роль GIL;
- подходы к автоматизированному тестированию Python-кода с использованием библиотеки pytest.

уметь:

- писать читаемый и идиоматичный Python-код, выбирать подходящие структуры данных и встроенные средства языка в соответствии с характером задачи;
- проектировать функции и классы с корректной сигнатурой, аннотациями типов;

- корректно обрабатывать исключения, работать с файлами, использовать контекстные менеджеры;
- реализовывать декораторы, итераторы и генераторы;
- писать автоматизированные тесты с использованием `pytest`.

владеть:

- инструментами разработки на Python;
- идиоматичными приёмами Python-разработки: распаковкой, срезами, списковыми включениями, встроенными функциями обработки коллекций, контекстными менеджерами;
- методами объектно-ориентированного проектирования: инкапсуляцией, наследованием, абстрактными интерфейсами и протоколами;
- инструментами автоматизированного тестирования на базе `pytest`;
- подходами к эффективной по памяти обработке данных с использованием генераторов и итераторов.

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-1.	Способен находить, формулировать и решать актуальные и значимые проблемы прикладной и компьютерной математики	ОПК-1.1.	Знает современные методы прикладной математики, а также алгоритмические основы решения задач в условиях реальных данных ретейла; владеет инструментами и технологиями, применяемыми для анализа и обработки больших объёмов информации (включая отечественные программные средства)
		ОПК-1.2.	Умеет формализовать бизнес-задачи ретейла в виде математических моделей, выбирать и применять алгоритмы и методы решения, интерпретировать результаты и представлять их в форме, доступной для технических и управленческих стейкхолдеров
		ОПК-1.3.	Имеет практический опыт решения прикладных задач в сфере ретейла в рамках учебных, научных или проектных работ, включая обработку реальных данных, разработку аналитических моделей и интеграцию решений в программные системы, а также участие в командных проектах, направленных на повышение эффективности бизнес-процессов
ОПК-3.	Способен самостоятельно создавать прикладные программные средства на основе современных информационных технологий и сетевых ресурсов, в том числе отечественного производства	ОПК-3.1.	Знает принципы проектирования и архитектуры программного обеспечения, современные языки программирования, фреймворки и инструменты разработки, а также особенности применения отечественных программных продуктов и платформ в условиях цифровой экономики
		ОПК-3.2.	Умеет разрабатывать, тестировать и сопровождать

			прикладные программные средства, интегрировать их с базами данных, аналитическими системами и внешними сервисами, а также адаптировать программные решения под требования бизнеса в сфере ретейла
		ОПК-3.3.	Имеет практический опыт разработки программных модулей или систем, используемых для реализации математических моделей и анализа данных в ретейле, включая участие в командных проектах, сопровождение кодовой базы, документирование решений и внедрение в производственную среду
ПК-6.	Способен представлять результаты профессиональной деятельности и формулировать практические рекомендации по развитию цифровых решений в ретейле	ПК-6.1.	Знает требования к подготовке технической и аналитической документации, принципы визуализации данных и стандарты профессиональной коммуникации
		ПК-6.2.	Умеет формулировать обоснованные выводы и рекомендации по результатам разработки и анализа, адаптируя их для технической и управленческой аудитории
		ПК-6.3.	Имеет практический опыт подготовки отчетов, презентаций и публичной защиты результатов проектов в области программной инженерии в ретейле

3. Тематический план

№ п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контроль	Самостояте льная работа	
		Лекции	Семинары (практичес кие занятия)			
1	Основы языка	4	4		17	Домашние задания Контроль теоретических знаний
2	Работа с данными и ошибками	4	4		17	Домашние задания Контроль теоретических знаний
3	Функции в Python	4	4		17	Домашние задания Контроль теоретических знаний
4	Организация кода	4	4		17	Домашние задания
5	Объектно- ориентированный Python	6	6		26	Домашние задания Контроль теоретических знаний
6	Итераторы, генераторы и устройство CPython	4	4		17	Домашние задания Контроль теоретических знаний
7	Конкурентное программирование	4	4		17	Домашние задания Контроль теоретических знаний
	Экзамен			2		
Итого:		30	30	2	128	
Объем дисциплины (модуля) (в ак. ч.)		190				
Объем дисциплины (модуля) (в зач. ед.)		5				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Основы языка	Python для тех, кто уже программирует Коллекции и pythonic-идиомы

2	Работа с данными и ошибками	Строки, байты, дата и время Обработка ошибок, файлы и контекстные менеджеры
3	Функции в Python	Функции Декораторы
4	Организация кода	Модули и пакеты
5	Объектно-ориентированный Python	Классы и объекты Dunder-методы и протоколы Дескрипторы и кастомизация доступа к атрибутам Наследование и полиморфизм
6	Итераторы, генераторы и устройство CPython	Генераторы и итераторы Модель памяти и внутреннее устройство CPython
7	Конкурентное программирование	Многопоточность и многопроцессность Асинхронное 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. Рамальо, Л. Python – к вершинам мастерства: лаконичное и эффективное программирование : практическое руководство / Л. Рамальо ; пер. с англ. А. А. Слинкина. — 2-е изд. — Москва : ДМК Пресс, 2022. — 900 с. — ISBN 978-5-97060-885-2. — Текст : электронный. — URL: <https://znanium.ru/catalog/product/2109578>.

2. Фаулер, М. Asyncio и конкурентное программирование на Python : практическое руководство / М. Фаулер ; пер. с англ. А. А. Слинкина. — Москва : ДМК Пресс, 2023. — 400 с. — ISBN 978-5-93700-166-5. — Текст : электронный. — URL: <https://znanium.ru/catalog/product/2109515>.

3. Чернышев, С. А. Основы программирования на Python : учебное пособие для вузов / С. А. Чернышев. — Москва : Издательство Юрайт, 2021. — 286 с. — (Высшее образование). — ISBN 978-5-534-14350-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/477353>.

Дополнительная литература:

1. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для вузов / Д. Ю. Федоров. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2021. — 210 с. — (Высшее образование). — ISBN 978-5-534-14638-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/478098>.

2. Чернышев, С. А. Принципы, паттерны и методологии разработки программного обеспечения : учебное пособие для вузов / С. А. Чернышев. — Москва : Издательство Юрайт, 2022. — 176 с. — (Высшее образование). — ISBN 978-5-534-14383-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/497029>.

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.	Научная электронная библиотека elibrary.ru библиотека	https://elibrary.ru/defaultx.asp
2.	База данных для IT-специалистов	https://habr.com
3.	База данных ScienceDirect	https://www.sciencedirect.com
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
5.	Федеральный портал «Российское образование»	https://www.edu.ru/
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
7.	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
8.	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
Среды разработки:		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое
Anaconda	зарубежное	свободно распространяемое
Robotic Operating System	зарубежное	свободно распространяемое
CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
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) «Язык программирования Python» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, домашние задания, контроль теоретических знаний, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Лекция – систематическое, последовательное, монологическое изложение преподавателем учебного материала, как правило, теоретического характера.

В процессе лекций рекомендуется вести конспект лекций: кратко и схематично фиксировать основные идеи, выводы и обобщения лекции; выделять важные мысли, ключевые слова и термины. Необходимо отметить вопросы или материалы, которые вызывают затруднения, и попытаться найти ответы в рекомендованной литературе. Если разобраться в материале не удастся, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

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.

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.

Контроль теоретических знаний – устные ответы на вопросы, список которых известен студенту заранее.

В процессе подготовки студенту необходимо проанализировать учебные материалы, ознакомившись с лекциями, учебниками и дополнительными источниками, акцентируя внимание на ключевых темах. Рекомендуется создать структурированные конспекты, выделяя основные идеи, термины и формулы.

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) «Язык программирования Python»

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

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме **экзамена**, 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, если иное не указано в силлабусе дисциплины:

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	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	
8	Отлично	
7	Хорошо	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы,
6	Хорошо	

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
		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 сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
5	Удовлетворительно	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	
3	Не сдан	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	
1	Не сдан	

Дисциплина (модуль) «Язык программирования Python» оценивается следующим образом:

Активность	Вес	Количество	Описание
Домашние задания	36%	6	Набор задач по темам недели
Контроль теоретических знаний	24%	4	Устный опрос по основным разделам и (или) темам дисциплины (модуля)
Экзамен	40%	1	Письменная или устная работа над заданием, направленным на проверку полученных знаний и навыков по дисциплине (модулю)

Формула расчёта итоговой оценки по дисциплине (модулю) «Язык программирования Python»: $0,36 \times \text{среднее за домашние задания} + 0,24 \times \text{контроль теоретических знаний} + 0,4 \times \text{экзамен}$.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание 1. Основы языка и работа с данными

1. Напишите функцию `process_names(names)`, которая:
 - Принимает список имён,
 - Удаляет дубликаты,
 - Приводит к нижнему регистру,
 - Возвращает отсортированный список с помощью спискового включения.
2. Напишите функцию `read_file_safely(path)`, которая читает файл с использованием `with`, обрабатывает `FileNotFoundError` и возвращает содержимое.
3. Создайте словарь с помощью генератора: ключи — числа от 1 до 10, значения — их

квадраты.

4. Оформите код в модуль, добавьте docstring и аннотации типов.

Домашнее задание 2. Функции, классы, декораторы

1. Напишите декоратор @timer, который выводит время выполнения функции. Примените его к функции, вычисляющей сумму чисел от 1 до 100 000.
2. Создайте класс Point с координатами x, y. Реализуйте:
 - __init__, __repr__, __eq__, __add__
 - Метод distance_to(origin)
3. Напишите протокол Drawable, и реализуйте его в классе Circle.
4. Напишите функцию, которая принимает список объектов Drawable и вызывает у каждого draw().

Домашнее задание 3. Итераторы, генераторы, асинхронность

1. Напишите генератор fibonacci(n), возвращающий первые n чисел Фибоначчи.
2. Напишите итератор Reverse, который обходит список в обратном порядке.
3. Напишите асинхронную функцию fetch_urls(urls), которая:
 - Использует aiohttp для скачивания страниц,
 - Возвращает список статус-кодов.
4. Запустите код с asyncio.run().
5. Оцените: почему генератор потребляет меньше памяти, чем список?

Вопросы для подготовки к контролю теоретических знаний

Тема 1: Основы языка и работа с данными

1. В чём разница между списком (list) и кортежем (tuple) в Python?
2. Как работает распаковка с * и ** в присваиваниях и вызовах функций?
3. Что такое списковое включение (list comprehension)? Приведите пример.
4. Какие типы данных в Python являются неизменяемыми (immutable)?
5. В чём разница между == и is?
6. Как объединить два словаря в Python 3.9+?
7. Что делает функция zip()? Как её использовать для обхода нескольких списков?
8. Как работает функция map()? Приведите пример её использования.
9. Что такое PEP 8 и зачем он нужен?
10. Как создать копию списка? В чём разница между поверхностной и глубокой копией?

Тема 2: Функции, декораторы, классы и ООП

1. Как определяется функция в Python? Какие виды параметров она может принимать?
2. Что такое аннотации типов? Как они используются в сигнатуре функции?
3. Что такое декоратор? Как создать простой декоратор, логирующий вызов функции?
4. В чём разница между @staticmethod и @classmethod?
5. Что такое дандер-методы? Приведите примеры __str__, __repr__, __eq__.
6. Как работает наследование в Python? Что такое MRO (Method Resolution Order)?
7. Что такое протокол в Python? Чем он отличается от интерфейса в других языках?
8. Как создать абстрактный метод с помощью ABC?
9. Что такое дескриптор? Какие методы он должен реализовать?
10. Как работает super() в наследовании? Зачем он нужен?

Тема 3: Итераторы, генераторы, асинхронность и внутреннее устройство

1. Что такое итератор? Какие методы он должен реализовать?
2. Что такое генератор? Чем он отличается от обычной функции?
3. Как работает ключевое слово yield? Приведите пример генератора.
4. В чём разница между return и yield внутри функции?

5. Что такое контекстный менеджер? Как создать его с помощью `@contextmanager`?
6. Что такое GIL (Global Interpreter Lock)? Как он влияет на многопоточность?
7. В чём разница между `threading` и `multiprocessing`?
8. Что такое асинхронное программирование? Какие ключевые слова используются в `asyncio`?
9. Как работает событийный цикл (event loop) в `asyncio`?
10. Почему генераторы потребляют меньше памяти, чем списки? Приведите пример использования.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Какой тип данных в Python подходит для хранения упорядоченной изменяемой последовательности элементов?	список (list)	ОПК-1
2.	Каким способом можно создать новый список, применив выражение к каждому элементу другого списка?	list comprehension	ОПК-3
3.	Какой тип коллекции в Python не поддерживает изменение после создания и может быть ключом в словаре?	кортеж (tuple)	ПК-6
4.	Какой оператор используется для перехвата исключения при выполнении потенциально опасного кода?	try-except	ОПК-1
5.	Какой контекстный менеджер обеспечивает автоматическое закрытие файла после работы с ним?	with	ОПК-3
6.	Какой модуль нужно импортировать, чтобы работать с датами и временем в Python?	datetime	ПК-6
7.	Как называется функция, которая принимает другую функцию в качестве аргумента или возвращает её?	функция высшего порядка	ОПК-1
8.	Какой символ используется для применения декоратора к функции?	@	ОПК-3
9.	Что создаётся при использовании ключевого слова <code>lambda</code> в Python?	анонимная функция	ПК-6
10.	Какой файл делает каталог распознаваемым как пакет при импорте?	init .py	ОПК-1
11.	Какие методы в Python определяют поведение объекта при сложении, сравнении или выводе?	dunder-методы	ОПК-3
12.	Какой протокол поддерживает объект, если в нём реализован метод <code>__next__</code> ?	итератор	ПК-6
13.	Какой механизм позволяет контролировать чтение, запись и удаление атрибутов класса?	дескриптор	ОПК-1
14.	Какой принцип ООП позволяет методу в подклассе переопределять поведение родительского класса?	полиморфизм	ОПК-3
15.	Что возвращает функция, содержащая <code>yield</code> , вместо <code>return</code> ?	генератор	ПК-6

16.	Какой модуль Python следует использовать для параллельного выполнения CPU-ёмких задач?	multiprocessing	ОПК-1
17.	Какой модуль подходит для выполнения I/O-операций в параллельных потоках?	threading	ОПК-3
18.	Как объявить асинхронную функцию, которую можно ожидать с помощью await?	async def	ПК-6
19.	Какой компонент управляет выполнением асинхронных задач в Python?	event loop	ОПК-1
20.	Какой принцип позволяет организовать код в отдельные файлы и импортировать их по необходимости?	модульность	ОПК-3