
УТВЕРЖДЕНА

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

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

Направление подготовки: 38.03.05 Бизнес-информатика

Направленность (профиль) подготовки: Бизнес и аналитика

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Срок освоения программы: 4 года

Год набора: 2026

**Москва
2026**

Содержание

1. Краткая характеристика дисциплины (модуля)	3
2. Перечень планируемых результатов обучения	5
3. Тематический план	7
4. Содержание дисциплины (модуля)	7
5. Учебно-методическое обеспечение	8
6. Материально-техническое обеспечение	8
7. Методические и оценочные материалы	10

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Продвинутый Python» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по направлению подготовки 38.03.05 Бизнес-информатика, профиль Бизнес и аналитика, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 838 от 29.07.2020 года.

Изучение дисциплины (модуля) «Продвинутый Python» позволяет писать качественный, масштабируемый и поддерживаемый код, соответствующий промышленным стандартам разработки. Python является ключевым инструментом в разработке приложений, анализе данных, автоматизации и интеграции систем.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки бакалавриата по направлению 38.03.05 Бизнес-информатика, профиль Бизнес и аналитика и входит в часть, формируемую участниками образовательных отношений, Блока 1, как дисциплина по выбору.

Дисциплина (модуль) изучается на 3 или 4 курсе в 6,7 или 8 семестре на выбор, доступна для изучения после успешного освоения дисциплины (модуля) «Язык программирования Python. Базовый» или «Язык программирования Python. Основной» либо дисциплины (модуля) «Язык программирования Java», «Язык программирования C++» или «Язык программирования Go».

Цель изучения дисциплины (модуля): заключается в формировании систематических навыков разработки структурированного, надёжного и поддерживаемого Python-кода с использованием продвинутых языковых возможностей и современных практик программирования.

Задачи изучения дисциплины (модуля):

- освоение основных и продвинутых конструкций языка Python для эффективной обработки данных и проектирования программ;
- развитие понимания принципов функционального и объектно-ориентированного программирования в контексте реальных задач;
- формирование и развитие способностей проектировать модульный, типизированный и устойчивый к ошибкам код;
- приобретение навыков работы с внешними данными, API и форматами обмена, а также организации кода в полноценные проекты;
- развитие компетенций в написании читаемого, документированного и переиспользуемого кода с применением современных инструментов разработки.

В результате освоения дисциплины (модуля), обучающийся должен:

знать:

- основные конструкции Python и принципы работы с коллекциями, вложенными структурами данных и файлами;
- принципы проектирования функций: параметры, возвращаемые значения, *args, **kwargs, замыкания и чистые функции;
- механизмы обработки ошибок и исключений: try, except, else, finally, raise, пользовательские исключения;
- принципы работы генераторов, генераторных выражений и ленивых вычислений;
- назначение декораторов, их синтаксис и области применения;
- принципы организации кода в модули, пакеты и проекты;
- основы работы с библиотеками, зависимостями, виртуальными окружениями и requirements.txt;

- назначение аннотаций типов, docstring, линтеров, форматтеров и принципов читаемого кода;
- основные принципы объектно-ориентированного программирования: классы, объекты, инкапсуляция, композиция, наследование;
- назначение моделей данных, dataclass, Pydantic и валидации входных данных;
- основы взаимодействия с внешними источниками данных, API, HTTP-запросами и форматами JSON/CSV.

уметь:

- использовать базовые и продвинутые конструкции Python для обработки данных;
- выбирать подходящие структуры данных для решения прикладных задач;
- проектировать функции с понятной сигнатурой и предсказуемым поведением;
- обрабатывать ошибки выполнения и корректно использовать исключения;
- создавать генераторы и применять их для потоковой обработки данных;
- разрабатывать и применять декораторы для вынесения повторяющейся технической логики;
- организовывать код в модули и пакеты, выстраивать структуру небольшого проекта;
- подключать стандартные и внешние библиотеки, управлять зависимостями проекта;
- добавлять аннотации типов, писать документацию к функциям и улучшать читаемость кода;
- проектировать классы, определять их ответственность, реализовывать взаимодействие объектов;
- использовать наследование, композицию и инкапсуляцию при проектировании программ;
- описывать модели данных с помощью dataclass и Pydantic;
- выполнять HTTP-запросы, обрабатывать ответы API и данные в форматах JSON/CSV.

владеть:

- навыками разработки структурированных Python-программ среднего уровня сложности;
- навыками проектирования функций, модулей и классов для решения прикладных задач;
- навыками обработки коллекций, вложенных структур, файлов и внешних данных;
- навыками построения устойчивого кода с обработкой ошибок и валидацией данных;
- навыками применения генераторов и декораторов для повышения эффективности и переиспользуемости кода;
- навыками организации проекта с использованием модулей, пакетов, виртуального окружения и зависимостей;
- навыками написания читаемого, типизированного и поддерживаемого кода;
- навыками объектно-ориентированного проектирования с использованием инкапсуляции, композиции и наследования;
- навыками моделирования данных и проверки их корректности;
- навыками интеграции Python-программ с внешними сервисами и API;
- навыками анализа, отладки и улучшения существующего программного кода.

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)
УК-1.	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1.	Знает методы поиска и анализа информации в области разработки, основные принципы критической оценки источников информации и их релевантности.
		УК-1.2.	Умеет критически оценивать источники информации и синтезировать данные из различных источников для решения задач, применять системный подход к анализу и решению комплексных проблем.
		УК-1.3.	Имеет практический опыт работы с современными инструментами и технологиями для обработки информации, формулировании и структурировании задач на основе полученной информации.
ОПК-2.	Способен проводить исследование и анализ рынка информационных систем и информационно-коммуникационных технологий, выбирать рациональные решения для управления бизнесом	ОПК-2.1.	Знает основные тенденции и характеристики рынка информационных систем и информационно-коммуникационных технологий
		ОПК-2.2.	Умеет проводить исследование и анализ рыночной информации для оценки потребностей бизнеса и выбора оптимальных решений
		ОПК-2.3.	Имеет практический опыт в разработке и внедрении стратегий управления бизнесом на основе анализа рынка информационных технологий
ПК-2.	Способен использовать соответствующий математический аппарат и инструментальные средства для обработки, анализа и систематизации информации по теме исследования для решения задач профессиональной деятельности	ПК-2.1.	Знает основные математические методы и инструментальные средства, применяемые для обработки и анализа информации
		ПК-2.2.	Умеет эффективно использовать математический аппарат для систематизации данных и решения профессиональных задач
		ПК-2.3.	Имеет практический опыт работы с инструментами анализа информации в рамках исследовательских проектов

ПК-3.	Способен готовить научно-технические отчеты, презентации, научные публикации по результатам выполненных исследований	ПК-3.1.	Знает требования и стандарты оформления научно-технических отчетов, презентаций и публикаций
		ПК-3.2.	Умеет структурировать и представлять результаты исследований в ясной и доступной форме
		ПК-3.3.	Имеет практический опыт подготовки и публикации научных материалов, отражающих результаты выполненных исследований
ПК-8.	Способен под руководством специалиста более высокой категории осуществлять планирование и организацию проектной деятельности на основе стандартов управления проектами	ПК-8.1.	Знает принципы и стандарты управления проектами
		ПК-8.2.	Умеет разрабатывать планы и организовывать проектную деятельность в соответствии с установленными стандартами
		ПК-8.3.	Имеет практический опыт участия в проектной работе, включая планирование и координацию задач

3. Тематический план

№п/ п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контроль	Самостоятель ная работа	
		Лекции	Семинары			
1	Базовые конструкции и обработка данных		4		8	Домашнее задание
2	Функции и функциональные инструменты		20		42	Домашнее задание Контрольная работа
3	Организация проекта и качество кода		12		26	Домашнее задание
4	Объектно- ориентированное программирование		24		50	Домашнее задание Контрольная работа
	<i>Зачёт с оценкой</i>			4		
<i>Итого:</i>			<i>60</i>	<i>4</i>	<i>126</i>	
<i>Объем дисциплины (модуля) (в ак. ч.)</i>		<i>190</i>				
<i>Объем дисциплины (модуля) (в зач. ед.)</i>		<i>5</i>				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Базовые конструкции и обработка данных	Базовые программы на Python, коллекции, вложенные структуры и файлы
2	Функции и функциональные инструменты	Проектирование функций. Ошибки и исключения. Генераторы. yield и ленивые вычисления. Декораторы и переиспользование логики
3	Организация проекта и качество кода	Модули и организация кода. Библиотеки и зависимости. Типизация и читаемость
4	Объектно-ориентированное программирование	Основы объектно-ориентированного программирования. Проектирование классов: инкапсуляция, ответственность, композиция. Наследование, расширение поведения и магические методы. Модели данных, валидация и контракт данных. Прикладной Python

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, 2026. — 349 с. — (Высшее образование). — ISBN 978-5-534-17139-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/588667>.
2. Виафоре, П. Надежный Python : практическое руководство / П. Виафоре, М. Райтман. - Санкт-Петербург : БХВ-Петербург, 2023. - 352 с. - ISBN 978-5-9775-1174-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2123392>.

Дополнительная литература:

1. Бейдер, Д. Чистый Python. Тонкости программирования для профи : практическое руководство / Д. Бейдер. - Санкт-Петербург : Питер, 2021. - 288 с. - (Серия «Библиотека программиста»). - ISBN 978-5-4461-0803-9. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/1766370>.
2. Хиллард, Д. Секреты Python Pro / Д. Хиллард. - Санкт-Петербург : Питер, 2021. - 320 с. - ISBN 978-5-4461-1684-3. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2142837>.

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
Robot 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) «Продвинутый Python» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, домашние задания, контрольная работа, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив. Кроме того, студентам предоставляется возможность получения бонусных баллов за высокий уровень выполнения домашних заданий, превышающий стандартные требования, проявление инициативы и креативности.

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.

Контрольная работа – письменная работа с набором задач, которые нужно решить за ограниченное время.

Цель контрольной работы - получить специальные знания по одной или нескольким

темам дисциплины (модуля) и продемонстрировать навыки их практического применения.

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) «Продвинутый 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 **зачёта с оценкой**, 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. 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, 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
6	Хорошо	

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
		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) «Продвинутый Python» оценивается следующим образом:

Активность	Вес	Количество	Описание
Домашнее задание	30%	13	Набор задач по темам недели
Контрольная работа	35%	2	Письменная работа с набором задач, которые нужно решить за ограниченное время
Зачёт с оценкой	35%	1	Письменная или устная работа над заданием, направленным на проверку полученных знаний и навыков по дисциплине (модулю)

Формула расчёта итоговой оценки по дисциплине (модулю) «Продвинутый Python»: $\langle 0,3 \times \text{среднее за домашние задания} + 0,35 \times \text{среднее за контрольные работы} + 0,35 \times \text{зачёт с оценкой} \rangle$.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание 1.

Задание 1. Напишите программу на Python, которая загружает данные из JSON-файла (например, список пользователей с полями: id, name, email, orders). Обработайте вложенную структуру: каждый пользователь может иметь список заказов с product, price, date. Реализуйте функции для: подсчёта общего числа заказов; нахождения пользователя с наибольшей суммой покупок; фильтрации заказов по дате (например, за последнюю неделю).

Задание 2. Работайте с CSV-файлом, содержащим данные о продажах (столбцы: product, category, price, quantity, date). С помощью стандартных средств Python (без pandas) реализуйте: чтение файла и парсинг строк; группировку по категории и подсчёт общей выручки; запись результата в новый JSON-файл.

Задание 3. Создайте скрипт, который обрабатывает несколько файлов (JSON, CSV) из одной директории. Реализуйте: автоматическое определение формата файла по расширению; унифицированную структуру данных для хранения записей; объединение всех данных в единый список и сохранение в файл all_data.json.

Домашнее задание 2.

Задание 1. Реализуйте функцию `process_orders(data, **filters)`, которая принимает список заказов и произвольные фильтры (например, `min_price=100, category="electronics"`). Используйте `*args` и `**kwargs` для гибкой сигнатуры. Функция должна возвращать отфильтрованный генератор (с `yield`), чтобы не загружать все данные в память.

Задание 2. Создайте генератор `log_parser(filename)`, который построчно читает лог-файл и возвращает словари с полями: `timestamp`, `level`, `message`. Реализуйте обработку ошибок: если строка не соответствует формату — пропускайте её и выводите предупреждение. Добавьте пользовательское исключение `InvalidLogFormatError`.

Задание 3. Напишите декоратор `@retry(max_attempts=3, delay=1)`, который повторяет выполнение функции при возникновении исключения. Используйте его для функции, имитирующей HTTP-запрос (с `random.choice` — успех или ошибка). Декоратор должен выводить сообщение о каждой попытке и задерживаться на указанное время.

Домашнее задание 3.

Задание 1. Организуйте код из ДЗ 1 и 2 в виде модулей: `data_loader.py` — загрузка и парсинг файлов; `filters.py` — фильтрация и обработка; `utils.py` — вспомогательные функции и исключения. Создайте `main.py`, который импортирует модули и выполняет полный пайплайн обработки. Убедитесь, что структура проекта позволяет легко добавлять новые форматы.

Задание 2. Настройте виртуальное окружение и файл `requirements.txt`, включающий: `requests` — для будущих HTTP-запросов; `pydantic` — для валидации данных; `mypy`, `flake8`, `black` — для контроля качества кода. Добавьте аннотации типов ко всем функциям из `data_loader.py` и `filters.py`. Проверьте код с помощью `mypy` и `flake8`.

Задание 3. Напишите `docstring` в формате Google Style для каждой функции.
Пример:

```
def load_json_data(path: str) -> List[Dict]:
    """Загружает данные из JSON-файла.
    Args:
    path: путь к файлу.
    Returns:
    Список словарей с данными.
    Raises:
    FileNotFoundError: если файл не найден.
    JSONDecodeError: если файл повреждён.
    """
```

Убедитесь, что код соответствует PEP 8 и проходит проверку линтером.

Примерные задания для контрольной работы

1. Напишите функцию `multiply(a, b=1, *args, **kwargs)`, которая возвращает произведение всех переданных числовых аргументов (из `a`, `b`, `*args`). Параметр `**kwargs` должен игнорироваться. Пример: `multiply(2, 3, 4) → 24`.

2. Объясните, в чём разница между `*args` и `**kwargs` в сигнатуре функции. Приведите пример использования каждого.

3. Напишите генератор `fibonacci(n)`, который возвращает первые `n` чисел Фибоначчи по одному с помощью `yield`. Используйте его для вывода первых 10 чисел.
4. Что такое ленивые вычисления? В чём преимущество генератора перед списком при обработке большого объёма данных?
5. Напишите декоратор `@timing`, который выводит время выполнения функции. Примените его к функции `slow_function()`, которая делает `time.sleep(1)` и возвращает `True`.
6. Объясните, зачем нужны пользовательские исключения. Создайте исключение `DataProcessingError` и используйте его в блоке `try-except`, чтобы обработать ошибку при обработке некорректного JSON.
7. Создайте класс `Product` с атрибутами `name`, `price`. Реализуйте метод `str`, чтобы при выводе объекта печаталось: "Товар: <name>, цена: <price>".
8. Реализуйте класс `Cart`, который хранит список товаров (объекты `Product`). Добавьте методы: `add_item(product)`, `total()` — возвращает общую сумму, и `len()` — возвращает количество товаров.
9. Что такое инкапсуляция? Как в Python обозначаются защищённые и приватные атрибуты? Приведите пример.
10. Используя библиотеку `pydantic`, опишите модель `User`, в которой: `name` — строка, `age` — целое число от 0 до 120, `email` — корректный email. Попробуйте создать объект с некорректным возрастом и объясните, что произойдёт.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1	В контексте обработки данных укажите структуру Python, предназначенную для хранения упорядоченной изменяемой последовательности элементов.	список / list	УК-1
2	В разделе базовых конструкций укажите тип данных, используемый для хранения пар "ключ-значение" в Python.	словарь / dict	УК-1
3	При работе с вложенными структурами укажите способ доступа к значению во вложенном словаре по ключам.	квадратные скобки / квадратные скобки последовательно / dict[key1][key2]	УК-1
4	В системном подходе к чтению данных укажите встроенный модуль Python для работы с JSON-файлами.	json	УК-1
5	При определении задач обработки табличных данных укажите встроенный способ чтения CSV-файла без использования сторонних библиотек.	модуль csv / csv.reader	ОПК-2
6	В выборе оптимальных способов обработки данных укажите оператор, используемый для распаковки аргументов функции из списка или кортежа.	* / звёздочка	ОПК-2
7	При формулировке целей для переиспользования логики укажите ключевое слово для определения функции в Python.	def	ОПК-2
8	В реальных условиях разработки укажите конструкцию, позволяющую обрабатывать исключения при работе с файлами.	try-except / try-except-finally	ОПК-2
9	В прикладной математике программирования укажите ключевое слово, используемое в генераторе вместо return.	yield	ПК-2

10	При формулировке задачи повторного использования кода укажите механизм, позволяющий оборачивать функции для добавления дополнительного поведения.	декоратор / decorator	ПК-2
11	В анализе результатов выполнения программы укажите встроенный инструмент Python для измерения времени выполнения блока кода.	time.time() / time.perf_counter()	ПК-2
12	При решении задач организации кода укажите способ группировки функций и классов в отдельный файл для повторного использования.	модуль / module	ПК-2
13	В математической формулировке задач типизации укажите синтаксис добавления аннотации типа к параметру функции.	: тип / : str, : int и т.д.	ПК-3
14	При обосновании структуры проекта укажите способ создания пакета в Python.	наличие файла __init__.py / наличие py-файлов и init-файла	ПК-3
15	В анализе качества кода укажите инструмент для автоматического форматирования Python-кода в соответствии с PEP 8.	black / autopep8	ПК-3
16	При формулировке задачи валидации данных укажите библиотеку для описания моделей данных с валидацией и аннотациями типов.	Pydantic	ПК-3
17	В решении типовых задач ООП укажите принцип, при котором объект содержит ссылки на другие объекты как часть своей реализации.	композиция / composition	ПК-8
18	При использовании ИКТ в разработке укажите магический метод, вызываемый при создании экземпляра класса.	__init__	ПК-8
19	В профессиональной деятельности с классами укажите механизм, позволяющий одному классу наследовать атрибуты и методы другого.	наследование / inheritance	ПК-8
20	При интеграции моделей данных в проект укажите способ указания, что поле в классе должно быть необязательным.	Optional[тип] / = None / Union[тип, None]	ПК-8