
УТВЕРЖДЕНА

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

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

Направление подготовки: 02.03.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Математика и компьютерные науки

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)	8
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. Основной» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по направлению 02.03.01 «Математика и компьютерные науки», профиль «Математика и компьютерные науки», утвержденный приказом Министерства науки и высшего образования Российской Федерации № 807 от 23.08.2017 года.

Изучение дисциплины (модуля) дает умение программирования, используемым в различных областях, таких как веб-разработка, анализ данных и машинное обучение. Освоение Python позволяет эффективно решать сложные задачи в профессиональной деятельности.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки бакалавриата по направлению 02.03.01 «Математика и компьютерные науки», профиль «Математика и компьютерные науки» и входит в обязательную часть Блока 1, как дисциплина по выбору.

Дисциплина (модуль) изучается на 1 курсе в 1 семестре.

Цель изучения дисциплины (модуля): формирование у студентов навыков программирования на языке Python, включая разработку, отладку и оптимизацию приложений.

Задачи изучения дисциплины (модуля):

- освоение основ языка Python и принципы структурирования кода;
- понимание базовых принципов анализа и визуализации данных;
- структурирование решений с использованием функций и классов;
- выполнение базовых обработок и анализ данных;

В результате освоения дисциплины (модуля), обучающийся должен:

знать:

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)
ОПК-6.	Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-6.1.	Знает алгоритмы разработки, компьютерные программы, а также алгоритмы вычислительной математики
		ОПК-6.2.	Умеет разрабатывать математические программные продукты и комплексы с использованием современных технологий программирования
		ОПК-6.3.	Имеет практический опыт разработки интеллектуальных информационных систем для визуализации результатов исследований
ПК-3.	Способен использовать методы математического и алгоритмического моделирования при решении теоретических и прикладных задач	ПК-3.1.	Знает основные методы математического и алгоритмического моделирования, а также их применение для решения теоретических и прикладных задач
		ПК-3.2.	Умеет разрабатывать и применять математические модели и алгоритмы для решения различных задач, анализируя полученные результаты
		ПК-3.3.	Имеет практический опыт использования методов математического и алгоритмического моделирования в реальных проектах или исследованиях
ПК-4.	Способен под руководством специалиста более высокой категории разрабатывать программное обеспечение для решения прикладных задач в сфере	ПК-4.1.	Знает основные принципы разработки программного обеспечения и методы решения прикладных задач
		ПК-4.2.	Умеет применять языки программирования и инструменты разработки для создания программного обеспечения под руководством более опытного специалиста

		ПК-4.3.	Имеет практический опыт участия в проектах по разработке программного обеспечения, работая в команде под руководством специалиста более высокой категории
--	--	---------	---

3. Тематический план

№п/ п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контроль	Самостоятельная работа	
		Лекции	Семинары			
1	Функции и функциональные инструменты		22	12	38	Домашнее задание Контест Контрольная работа
2	Организация проекта и качество кода		12	6	22	Домашнее задание Контест
3	Объектно-ориентированное программирование		14	8	26	Домашнее задание Контест Контрольная работа
4	Прикладная разработка на Python		8	4	14	Домашнее задание Контест
	Зачет с оценкой			4		
Итого:			56	34	100	
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	Функции и функциональные инструменты	Базовые программы на Python, коллекции, вложенные структуры и файлы Проектирование функций Ошибки и исключения Генераторы, yield и ленивые вычисления Декораторы и переиспользование логики
2	Организация проекта и качество кода	Модули и организация кода Библиотеки и зависимости Типизация и читаемость
3	Объектно-ориентированное программирование	Основы объектно-ориентированного программирования Проектирование классов: инкапсуляция, ответственность, композиция Наследование, расширение поведения и магические методы
4	Прикладная разработка на Python	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, 2025. — 349 с. — (Высшее образование). — ISBN 978-5-534-17139-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/567821>.
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
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. Основной» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, коллоквиумы и проекты, 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.

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

Контекст 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.

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 и продемонстрировать навыки их практического применения.

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.

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 «Язык программирования Python. Основной»

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 *зачета 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	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. 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
9	Отлично	Зачтено	
8	Отлично	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			дисциплины (модуля) с практическими задачами.
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 сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
6	Хорошо	Зачтено	
5	Удовлетворительно	Зачтено	Студент обладает базовыми знаниями по дисциплине, 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	

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

Активность	Вес	Количество	Описание
Домашнее задание	15%	13	Устные ответы на вопросы, список которых известен студенту заранее
Контеcт	25%	13	Интерактивные задания разного уровня сложности с автоматической проверкой результатов

Активность	Вес	Количество	Описание
Контрольная работа	30%	2	Письменная работа с набором задач, которые нужно решить за ограниченное время
Зачет с оценкой	30%	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. Основной»: $0,15 \times \text{среднее за домашние задания} + 0,25 \times \text{среднее за контесты} + 0,3 \times \text{среднее за контрольные работы} + 0,3 \times \text{зачет с оценкой}$.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

1. Функции и модули

1. Напишите модуль `math_utils.py` с функциями:
 - `gcd(a, b)` — НОД двух чисел,
 - `lcm(a, b)` — НОК,
 - `is_prime(n)` — проверка на простоту.
2. Импортируйте модуль в основной файл и протестируйте функции.

2. Работа с файлами и форматами

1. Прочитайте текст из файла `input.txt`, подсчитайте количество слов, строк и символов.
2. Запишите результат в `stats.txt`.
3. Сохраните данные в формате JSON:

```
json
Свернуть
Копировать
{"lines": 10, "words": 150, "chars": 900}
```

1

3. Словари и коллекции

1. Создайте словарь с данными о студентах: имя, возраст, оценки (список).
2. Напишите функции:
 - `average_grade(student)` — средний балл студента,
 - `top_student(students)` — студент с лучшим средним баллом.
3. Выведите результат.

4. Исключения и обработка ошибок

1. Напишите программу, которая запрашивает число и возводит его в квадрат.
2. Обработайте:
 - `ValueError` — если введена не цифра,
 - `KeyboardInterrupt` — если пользователь нажал Ctrl+C.
3. Используйте `try-except-else-finally`.

5. Объектно-ориентированное программирование (ООП)

1. Создайте класс `BankAccount` с атрибутами:
 - `owner` (имя владельца),

- `balance` (баланс, по умолчанию 0).
- 2. Реализуйте методы:
 - `deposit(amount)`,
 - `withdraw(amount)`,
 - `__str__()` — строковое представление.
- 3. Создайте два счёта, внесите и снимите деньги, выведите информацию.

6. Работа с датой и временем

1. Напишите программу, которая:
 - Выводит текущую дату и время,
 - Определяет, сколько дней осталось до Нового года,
 - Проверяет, был ли введённый год високосным.
2. Используйте модуль `datetime`.

7. Работа с JSON и API (на примере)

1. Создайте словарь с данными о книгах: название, автор, год.
2. Сохраните в `books.json`.
3. Прочитайте файл и выведите все книги, изданные после 2000 года.

8. Генераторы и итераторы

1. Напишите генератор `fibonacci(n)`, возвращающий первые `n` чисел Фибоначчи.
2. Напишите генератор `even_numbers(start, end)`, возвращающий чётные числа в диапазоне.
3. Используйте `yield`.

9. Декораторы

1. Напишите декоратор `@timer`, который измеряет время выполнения функции.
2. Примените его к функции `slow_function()`, которая спит 2 секунды (`time.sleep(2)`).
3. Выведите: `Функция выполнялась за 2.03 сек.`

10. Итоговый проект: "Система учёта задач (To-Do List)"

Реализуйте консольное приложение с функциями:

- Добавление задачи (название, приоритет, статус).
- Просмотр всех задач.
- Фильтрация по статусу (выполнено/не выполнено).
- Сохранение и загрузка из `tasks.json`.
- Использование классов, файлов, исключений, JSON.

Примерные задания для конкурса

№	ЗАДАНИЕ	ОГРАНИЧЕНИЯ	ПРИМЕР
1	Сумма простых чисел до N	$1 \leq N \leq 1000$	Ввод: 10 → Вывод: 17 (2+3+5+7)

№	ЗАДАНИЕ	ОГРАНИЧЕНИЯ	ПРИМЕР
2	Наибольшая подстрока без повторов	Длина строки ≤ 100	Ввод: <code>abcabcbb</code> → Вывод: <code>3 (abc)</code>
3	Слияние словарей с суммированием	Два словаря	Ввод: <code>{'a': 1, 'b': 2}, {'b': 3, 'c': 1}</code> → Вывод: <code>{'a': 1, 'b': 5, 'c': 1}</code>
4	Палиндром с учётом регистра и пробелов	Строка ≤ 100 символов	Ввод: <code>A man a plan a canal Panama</code> → Вывод: <code>Да</code>
5	Разложение числа на простые множители	$2 \leq N \leq 1000$	Ввод: <code>12</code> → Вывод: <code>[2, 2, 3]</code>
6	Самая длинная общая подпоследовательность (LCS)	Две строки ≤ 50	Ввод: <code>abcde, ace</code> → Вывод: <code>3 (ace)</code>
7	Генерация всех перестановок	Строка из 3–6 символов	Ввод: <code>abc</code> → Вывод: <code>['abc', 'acb', ...]</code>
8	Проверка валидности email	Только латиница, <code>@</code> , <code>.</code>	Ввод: <code>test@mail.ru</code> → Вывод: <code>Да</code>
9	Подсчёт слов в тексте с сортировкой по частоте	Текст до 1000 слов	Ввод: <code>hello world hello</code> → Вывод: <code>[('hello', 2), ('world', 1)]</code>
10	Реализация стека с проверкой скобок	Использовать список как стек	Ввод: <code>{(){} }</code> → Вывод: <code>Да</code>

Примерные задания для контрольных работ

Контрольная работа №1 (Темы: функции, исключения, файлы)

Время: 60 минут

- Напишите функцию `read_file(filename)`, которая читает файл и возвращает его содержимое. Обработайте `FileNotFoundError`.
- Напишите функцию `factorial(n)`, выбрасывающую `ValueError`, если `n < 0`.
- Что выведет код?

```
def func(x, y=[]):
    y.append(x)
    return y
print(func(1))
print(func(2))
```
- Запишите словарь `{'name': 'Alice', 'age': 25}` в файл `user.json`.
- Найдите ошибку:

python

Свернуть

Копировать

```
1 data = json.load("data.txt")
```

Ответы:

3. [1], [1, 2] (из-за мутабельного аргумента по умолчанию)

5. json.load() принимает файловый объект, а не строку → нужно open("data.txt")

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1	Назовите основную HTTP-метод, используемую для получения данных.	GET / POST / PUT	ПК-4
2	Укажите основную команду Git для добавления файлов в индекс.	add / commit / push	ОПК-6
3	Назовите уровень в архитектуре REST API, отвечающий за ресурсы.	ресурсы / представление / контроллер	ПК-3
4	Укажите основную функцию библиотеки requests для отправки данных на сервер.	post / get / put	ПК-4
5	Студент, назовите протокол для обмена данными в веб-разработке, используемый для REST API.	HTTP / HTTPS / FTP	ПК-4
6	Укажите принцип модульного кода, обеспечивающий разделение логики.	разделение логики / инкапсуляция / абстракция	ОПК-6
7	Назовите компонент клиент-серверной архитектуры, отвечающий за хранение данных.	сервер / база данных / клиент	ПК-4
8	Укажите тип данных в JSON, представляющий список.	array / object / string	ПК-3
9	Назовите этап работы с FastAPI, связанный с созданием эндпоинтов.	эндпоинты / установка / тестирование	ОПК-6
10	Укажите способ интеграции HTML-шаблонов в FastAPI, использующий Jinja2.	Jinja2 / встроенный / внешний	ПК-4
11	Назовите тип ошибки в веб-приложениях, связанный с отсутствием ресурса.	404 / 500 / 403	ПК-3
12	Укажите метод обработки форм в FastAPI для получения данных из POST-запроса.	POST / GET / PUT	ПК-4
13	Назовите принцип разработки ПО, обеспечивающий модульность.	модульность / инкапсуляция / наследование	ОПК-6
14	Укажите инструмент для работы с GitLab, для управления ветками.	ветки / репозитории / merge	ОПК-6
15	Назовите шаг для настройки API в FastAPI, связанный с валидацией данных.	валидация / маршрутизация / документация	ПК-3
16	Укажите способ обработки ошибок в FastAPI, использующий исключения.	исключения / коды / логи	ПК-3
17	Назовите этап разработки бота с инлайн-кнопками, для тестирования обработчиков.	тестирование / разработка / интеграция	ПК-4
18	Укажите метод обеспечения стабильности веб-приложения, связанный с логированием.	логирование / мониторинг / резервное копирование	ПК-3

