УТВЕРЖДЕНО

ООО«Учи.Дома»

Маршеня В. В.

04.08.2026

1. Сведения об организации

№ п/п	Наименование поля	Значение поля
1.1	**Наименование образовательной организации**	ООО «Учи.Дома»
1.2	**Логотип образовательной организации**	
1.3	**ИНН**	9704034757
1.4	**Ответственный за реализацию Комплекта ДОП, ФИО**	Маршеня Виктория Вадимовна
1.5	**Ответственный за реализацию Комплекта ДОП, должность**	Руководитель направления
1.6	**Ответственный за реализацию Комплекта ДОП, телефон**	+79505638553
1.7	**Ответственный за реализацию Комплекта ДОП, электронная почта**	v.marchenya@vkteam.ru

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

2. Комплекс основных характеристик ДОП Комплекта ДОП

Основные данные Комплекта ДОП

№ п/п	Наименование поля	Значение поля
1.	Направление обучения	Современные языки программирования
2.	Название Комплекта ДОП	Программирование на Python (Школьники 8–9 классов) Онлайн
3.	Название каждой ДОП Комплекта ДОП	Python Старт (базовый уровень, 8–9 классы) Python Про (продвинутый уровень, 8–9 классы)
4.	Вид ДОП входящих в Комплект ДОП	Дополнительные общеобразовательные программы
5.	Ссылка на Курс	https://futurecode.uchi.ru/python-8-9-online/
6.	Форма обучения	Очная
7.	Формат обучения	Онлайн
8.	Целевая аудитория ДОП	Школьники 8–9 классов
9.	Язык программирования	Python
10.	Особенности содержания ДОП по современным языкам программирования	Комплект ДОП нацелен на изучение современных языков программирования

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Аннотация Комплекта ДОП

№ п/п	Наименование поля	Значение поля
1.	Аннотация Комплекта ДОП	1) сведения о направленности представленных ДОП и характеристиках Целевой аудитории ДОП Дополнительные общеобразовательные программы «Python Старт (базовый уровень, 8–9 классы)» и «Python Про (продвинутый уровень, 8–9 классы)» имеют техническую направленность и относятся к направлению обучения «Современные языки программирования». Программы ориентированы на обучающихся 8–9 классов общеобразовательных организаций и разработаны с учетом их возрастных особенностей, уровня подготовки, познавательных интересов и образовательных потребностей. Содержание программ направлено на развитие алгоритмического мышления, цифровой грамотности, навыков программирования и решения практических задач средствами языка Python. 2) краткое описание цели и содержания каждой ДОП Python Старт (базовый уровень, 8–9 классы) Цель программы — формирование у обучающихся 8–9 классов специализированных знаний и практических навыков программирования на языке Python, развитие алгоритмического мышления, цифровой грамотности, навыков ответственного использования ИИ-технологий и мотивации к дальнейшему обучению в сфере информационных технологий через решение практических задач и выполнение учебных проектов. Программа направлена на изучение основных алгоритмических конструкций, типов данных и базовых возможностей языка программирования Python, выполнение практических работ по созданию алгоритмов и решению классических задач, а также формирование первичного опыта разработки программных решений посредством выполнения индивидуальных и командных учебных проектов с учетом возрастных особенностей обучающихся, их уровня подготовки и образовательных потребностей. Содержание: Изучение базовых конструкций языка Python: переменных, типов данных, операторов, условных конструкций, циклов, строк, коллекций, функций и модулей. Освоение основ алгоритмизации и программирования через решение типовых задач. Выполнение практических работ по созданию консольных программ и обработке строк и коллекций данных. Знакомство с принципами объектно-ориентированного программирования на примере создания графических приложений с использованием библиотеки Turtle. Использование ИИ-инструментов в программировании. Выполнение индивидуальных и командных учебных проектов. Решение кейс-задачи — выполнение практического задания по решению проблемной ситуации в рамках направления обучения. Python Про (продвинутый уровень, 8–9 классы) Цель программы — формирование у обучающихся 8–9 классов углубленных знаний и практических навыков программирования
		на языке Python, совершенствование алгоритмического мышления, развитие цифровых компетенций, навыков ответственного использования ИИ-технологий и формирование мотивации к дальнейшему обучению и профессиональному развитию в сфере информационных технологий через решение практико-ориентированных задач повышенной сложности и выполнение проектов. Программа направлена на углубленное изучение языка программирования Python, объектно-ориентированного программирования, алгоритмов и структур данных, современных технологий разработки программного обеспечения, развитие навыков проектной деятельности посредством выполнения индивидуальных и командных проектов, а также подготовку к участию в олимпиадах, конкурсах и соревнованиях в сфере информационных технологий с учетом возрастных особенностей обучающихся, их уровня подготовки и образовательных потребностей. Содержание: Углублённое изучение языка Python, объектно-ориентированного программирования, структур данных и алгоритмов. Освоение функций, модулей и принципов организации программ. Изучение и практическое применение алгоритмов поиска и сортировки, а также базовых структур данных. Работа с файлами, форматами JSON и CSV, основами внешних библиотек и обработкой данных. Изучение основ веб-взаимодействия и API на базовом уровне, а также принципов получения и обработки данных из внешних источников. Решение задач повышенной сложности, направленных на развитие алгоритмического мышления. Использование ИИ-инструментов в программировании. Выполнение индивидуальных и командных учебных проектов. Решение кейс-задачи — выполнение практического задания по решению проблемной ситуации в рамках направления обучения. 3) описание требований и рекомендаций для обучения по ДОП; Python Старт (базовый уровень, 8–9 классы) Специализированные знания и навыки программирования не требуются. Требуется наличие знаний школьной программы по математике на уровне не ниже 7 класса, навыков работы с персональным компьютером, логического мышления, общей цифровой грамотности и знание базовой терминологии в области алгоритмизации и программирования. Программа ориентирована на выполнение практических работ по созданию алгоритмов и решению классических задач программирования, а также разработку учебного проекта. Python Про (продвинутый уровень, 8–9 классы) Требуются специализированные знания и навыки программирования на языке Python. Предполагается наличие знаний школьной программы по математике и информатике на уровне не ниже 7 класса, понимание основных конструкций

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

		языка Python, опыт решения практических задач, развитое логическое мышление, навыки самостоятельной работы с учебными материалами и общая цифровая грамотность. Программа ориентирована на углубленное изучение языка Python, выполнение практико-ориентированных заданий и проектов, а также подготовку к участию в профильных олимпиадах, конкурсах и соревнованиях. 4) краткое описание результатов обучения в свободной форме, включая описание практико-ориентированного характера каждой ДОП Python Старт (базовый уровень, 8–9 классы) По итогам освоения программы обучающиеся будут знать основные конструкции языка Python, понимать принципы создания, тестирования и отладки программ, использовать переменные, условные конструкции, циклы, функции, строки и коллекции при решении практических задач, познакомятся с базовыми принципами объектно-ориентированного программирования на примере разработки графических приложений с использованием библиотеки Turtle, а также приобретут навыки поиска и исправления ошибок в программном коде, ответственного использования технологий искусственного интеллекта и выполнения индивидуальных и командных учебных проектов. Программа носит практико-ориентированный характер и предусматривает последовательное изучение основных алгоритмических конструкций, выполнение практических работ по созданию алгоритмов и решению классических задач программирования, а также формирование опыта разработки программных решений посредством выполнения индивидуальных и командных учебных проектов с учетом возрастных особенностей обучающихся, их уровня подготовки и образовательных потребностей. Python Про (продвинутый уровень, 8–9 классы) По итогам освоения программы обучающиеся будут знать язык программирования Python, объектно-ориентированное программирование, алгоритмы и структуры данных, понимать принципы создания, тестирования и отладки программ, применять современные технологии разработки программного обеспечения, включая работу с файлами, API, форматами JSON и CSV, регулярными выражениями, тестированием и системами контроля версий, а также приобретут навыки поиска, анализа и исправления ошибок в программном коде, ответственного использования технологий искусственного интеллекта и выполнения индивидуальных и командных проектов. Программа носит практико-ориентированный характер и предусматривает углубленное изучение языка программирования Python, объектно-ориентированного программирования, алгоритмов и структур данных, современных технологий разработки программного обеспечения,
		развитие навыков проектной деятельности посредством выполнения индивидуальных и командных проектов, а также подготовку к участию в олимпиадах, конкурсах и соревнованиях в сфере информационных технологий с учетом возрастных особенностей обучающихся, их уровня подготовки и образовательных потребностей.

Основные данные ДОП: БАЗОВЫЙ УРОВЕНЬ

№ п/п	Наименование поля	Значение поля
1.	Название ДОП	Python Старт (базовый уровень, 8–9 классы)
2.	Уровень обучения ДОП	БАЗОВЫЙ
3.	Объем/общая трудоемкость ДОП (общий объем освоения ДОП в академических часах)	146.00
4.	Объем каждого модуля в ак.ч.	Модуль1 - 36 Модуль2 - 36 Модуль3 - 36 Модуль4 - 38
5.	Объем недельной нагрузки обучающегося	4.00
6.	Дата начала и дата окончания обучения по ДОП (срок освоения ДОП)	С 2026-09-28 по 2027-05-31

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

7.	Кейс-задача	Название Кейс-задачи: Пароль, который можно запомнить Описание проблемной ситуации: Пароли используются для входа в социальные сети, электронную почту, учебные платформы, онлайн-магазины и другие цифровые сервисы. При этом многие пользователи продолжают использовать простые пароли, которые легко подобрать, либо используют один и тот же пароль сразу для нескольких сервисов. Сложные случайные пароли обычно надёжнее, но их трудно запомнить. Из-за этого возникает противоречие: пароль должен быть безопасным, но при этом удобным для пользователя. Цель решения: Разработать командное решение, которое поможет пользователям создавать, проверять или улучшать пароли так, чтобы они были более надёжными и при этом оставались удобными для запоминания или использования. Команда самостоятельно определяет способ достижения результата и состав разрабатываемого продукта. Контекст: Защита учётных записей является одной из важных задач информационной безопасности. Пользователи ежедневно работают с большим количеством сервисов и аккаунтов, для доступа к которым используются пароли. При этом требования к надёжности паролей постоянно растут, а количество утечек данных и попыток несанкционированного доступа остаётся высоким. Для решения этой проблемы используются различные подходы: генераторы паролей, парольные фразы, менеджеры паролей, системы оценки надёжности паролей и средства автоматической проверки соблюдения требований безопасности. Несмотря на большое количество существующих решений, поиск баланса между безопасностью и удобством использования паролей остаётся актуальной задачей. Вопросы и задания: Какую проблему вы выбрали для решения и почему считаете её важной? Какую цель вы поставили перед своим проектом и какого результата планируете достичь? Какое решение проблемы предложила ваша команда? Расскажите о распределении задач между участниками. Покажите результат работы и объясните, как он помогает решить выбранную проблему. Какие сильные стороны имеет ваше решение и какие его недостатки или ограничения вы можете выделить? Какие улучшения или дополнительные возможности можно реализовать в будущем? Возможный результат выполнения задания: Подготовлена защита кейс-задачи, включающая обоснование выбранной проблемы, описание предложенного решения, демонстрацию созданного прототипа, MVP или программного продукта, анализ сильных сторон и ограничений решения, обсуждение возможных направлений его развития и описание вклада участников команды в общий результат. Критерии оценки: Проблема сформулирована и обоснована — 1 балл. Цель и ожидаемый результат соответствуют выбранной проблеме — 1 балл. Предложено решение, направленное на устранение или уменьшение проблемы — 1 балл. Создан прототип, MVP или работоспособный программный продукт — 1 балл. Представлена демонстрация решения и объяснён принцип его работы — 1 балл. Проведён анализ решения: определены его сильные стороны, ограничения и возможные направления развития — 1 балл. Объяснено распределение
		задач между участниками команды и вклад каждого участника в общий результат — 1 балл. Каждый вопрос оценивается в 1 балл. Балл выставляется, если команда дала содержательный и аргументированный ответ на соответствующий вопрос и продемонстрировала результат работы. Кейс-задача считается выполненной успешно, если команда набрала не менее 5 баллов из 7 возможных. Ограничения и допущения: Решение должно быть реализовано с использованием языка программирования Python. При необходимости допускается использование сторонних библиотек, файлов, баз данных, графических интерфейсов и других инструментов, если они помогают решить поставленную задачу. Технические требования к оборудованию и программному обеспечению: Для решения кейс-задачи не требуется специализированное оборудование или программное обеспечение.
8.	Выдаваемый по окончании обучения документ	Все Получатели поддержки, успешно прошедшие промежуточную аттестацию по модулю ДОП, получают Сертификат Провайдера об освоении модуля. Все Получатели поддержки, успешно прошедшие итоговый контроль/аттестацию по ДОП допускаются к процедуре ИТиСИ. Все Получатели поддержки, успешно прошедшие ИТиСИ, получают Документ об обучении (форма «Университет 2035»).

Описание ДОП

№ п/п	Наименование поля	Значение поля

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

1.	Актуальность ДОП	В условиях стремительного развития цифровой экономики и активного внедрения информационных технологий во все сферы деятельности возрастает потребность в формировании у обучающихся современных цифровых компетенций, навыков программирования и разработки программного обеспечения. Одним из наиболее востребованных инструментов для их формирования является язык программирования Python, который сочетает простоту освоения, широкие возможности применения и востребованность в различных областях информационных технологий. Отличительными особенностями содержания программы являются практико-ориентированная направленность обучения, последовательное изучение основных алгоритмических конструкций, типов данных и базовых возможностей языка программирования Python, выполнение практических работ по созданию алгоритмов и решению классических задач, а также формирование первичного опыта разработки программных решений посредством выполнения индивидуальных и командных учебных проектов с учетом возрастных особенностей обучающихся, их уровня подготовки и образовательных потребностей. Новизна программы заключается в сочетании изучения языка программирования Python с использованием современных технологий искусственного интеллекта и инструментов разработки программного обеспечения. Программа способствует развитию алгоритмического мышления, цифровой грамотности, навыков ответственного использования ИИ-технологий и мотивации к дальнейшему обучению в сфере информационных технологий.
2.	Цель и задачи ДОП	Цель программы: Формирование у обучающихся 8–9 классов специализированных знаний и практических навыков программирования на языке Python, развитие алгоритмического мышления, цифровой грамотности, навыков ответственного использования ИИ-технологий и мотивации к дальнейшему обучению в сфере информационных технологий через решение практических задач и выполнение учебных проектов. Задачи программы: Обучающие: сформировать навыки создания, запуска, тестирования и отладки программ на языке Python; сформировать знания основных конструкций языка Python: переменных, типов данных, операторов, условных конструкций, циклов, строк, коллекций, функций и модулей; познакомить с базовыми принципами объектно-ориентированного программирования на примере создания графических приложений с использованием библиотеки Turtle; познакомить с возможностями использования технологий искусственного интеллекта в процессе разработки программ; сформировать основы ответственного и безопасного использования информационных технологий. Развивающие: развивать алгоритмическое, логическое и критическое мышление; развивать навыки поиска, анализа и исправления ошибок, а также отладки программ; развивать навыки самостоятельной работы с документацией и чтения программного кода; развивать навыки проектной, исследовательской и командной деятельности при выполнении индивидуальных и командных проектов; развивать способность применять современные технологии программирования для решения практических задач. Воспитательные: формировать ответственное отношение к работе с данными, включая вопросы конфиденциальности, безопасности и целостности информации; воспитывать этичное использование технологий искусственного интеллекта, уважение к авторским правам и интеллектуальной собственности; формировать навыки эффективной коммуникации и конструктивного взаимодействия при выполнении командных проектов; способствовать формированию устойчивого интереса к программированию, а также мотивации к дальнейшему обучению в сфере информационных технологий.

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

3.	Планируемые результаты	Планируемые результаты По итогам освоения программы обучающийся будет: Иметь представление: о возможностях языка программирования Python для разработки программ; о базовых принципах объектно-ориентированного программирования; о возможностях использования технологий искусственного интеллекта в процессе разработки программ; о принципах ответственного и безопасного использования информационных технологий. Знать: основные конструкции языка Python: переменные, типы данных, операторы, условные конструкции, циклы, строки, коллекции, функции и модули; основные принципы объектно-ориентированного программирования; основные возможности библиотеки Turtle для создания графических приложений. Понимать: принципы создания, запуска, тестирования и отладки программ на языке Python; принципы разработки графических приложений с использованием библиотеки Turtle; возможности и ограничения использования технологий искусственного интеллекта в процессе разработки программ; принципы ответственного и безопасного использования информационных технологий. Уметь: создавать, запускать, тестировать и отлаживать программы на языке Python; использовать основные конструкции языка Python при решении практических задач; разрабатывать графические приложения с использованием библиотеки Turtle; применять современные технологии программирования для решения практических задач; выполнять индивидуальные и командные учебные проекты. Владеть: навыками создания, запуска, тестирования и отладки программ на языке Python; навыками поиска, анализа и исправления ошибок в программном коде; навыками самостоятельной работы с документацией и чтения программного кода; навыками ответственного использования технологий искусственного интеллекта; навыками проектной, исследовательской и командной деятельности при выполнении индивидуальных и командных проектов.
4.	Дополнительная информация о ДОП	Дополнительная информация отсутствует
5.	Дополнительно (в соответствующих отдельных графах) указывается следующая информация по ДОП:	
5.1.	Стоимость обучения по ДОП	35558
5.2.	Рекомендации по обучению на ДОП	При обучении по программе рекомендуется руководствоваться дополнительной общеобразовательной программой, соблюдать последовательность освоения учебного материала, требования к организации обучения, выполнять предусмотренные программой формы текущего контроля, промежуточной и итоговой аттестации, а также использовать учебно-методические материалы.

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

3. Комплекс организационно-педагогических условий реализации ДОП:

Учебный план

№ п/п	Элементы ДОП	Элементы модуля (темы, промежуточная аттестация)	Общее кол-во часов элемента модуля, ак.час	Из них:			
				теоретическая подготовка, ак.час	практическая работа, ак.час	самостоятельная работа и самоконтроль, ак.час	контроль/ аттестация, ак.час
1.	Модуль 1 Первая программа	Введение в Python	17	1	10	5	1
2.		Условный оператор	7	0	4	2	1
3.		Использование ИИ-инструментов	2	0	1	1	0
4.		Цикл с условием	6	1	3	1	1
5.		Проект	3	0	2	1	0
6.		Промежуточная аттестация	1	0	0	0	1
7.	Модуль 2 Обработка строк	Строки	15	1	8	5	1
8.		Функции	8	1	4	2	1
9.		Цикл for	7	0	4	2	1
10.		Командный проект	5	0	4	1	0
11.		Промежуточная аттестация	1	0	0	0	1
12.	Модуль 3 Коллекции	Списки и кортежи	10	1	5	3	1
13.		Ответственное использование ИИ-инструментов	2	0	1	1	0
14.		Сортировка списков	8	1	4	2	1
15.		Словари и множества	12	0	8	3	1
16.		Проект	3	0	2	1	0
17.		Промежуточная аттестация	1	0	0	0	1
18.	Модуль 4 Модуль Turtle	Модули в Python	6	1	2	2	1
19.		Модуль Turtle	22	2	13	6	1
20.		Кейс-задача	7	0	5	2	0
21.		Промежуточная аттестация	1	0	0	0	1
Итоговый контроль/аттестация по ДОП			2.00	0	0	0	2.00
Итого по ДОП:			146	9	80	40	17
ИТиСИ			Объем часов ИТиСИ не входит в общую трудоемкость ДОП, но участие в ИТиСИ обязательно для всех успешно прошедших Итоговый контроль/аттестацию по ДОП				

Рабочая программа с описанием каждого модуля ДОП

Контур Крипто
владелец Маршеня Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Элементы ДОП	Элементы модуля	Содержание (единицы содержания теоретической подготовки, практической работы, самостоятельной работы, контроля по теме и промежуточной аттестации (вопросы, задания, задачи и пр.))	Виды образовательных мероприятий /деятельности (теоретическая подготовка, практическая работа, самостоятельная работа, аттестация/контроль)	Объем в ак.ч.
Модуль 1 Первая программа Модуль направлен на формирование у обучающихся базовых представлений о программировании и языке Python, освоение основных конструкций языка, развитие алгоритмического мышления и навыков решения практических задач. В рамках модуля обучающиеся знакомятся с образовательной платформой и средой разработки, изучают базовые типы данных, переменные, арифметические и логические операции, условные конструкции (if, elif, else), цикл while, операторы break и continue, а также получают первый опыт использования технологий искусственного интеллекта в программировании и выполняют проект «Калькулятор». Задачи модуля: познакомить обучающихся с образовательной платформой и средой программирования; сформировать знания основных конструкций языка Python; развить навыки работы с переменными, типами данных и операциями; сформировать навыки использования условных конструкций (if, elif, else); сформировать навыки использования цикла while и операторов break, continue; познакомить с возможностями использования технологий искусственного интеллекта при разработке программ; сформировать навыки разработки, тестирования и отладки простых программ; развить навыки проектной деятельности при выполнении индивидуального проекта. Планируемые результаты обучения По итогам освоения модуля обучающийся будет: знать: основные элементы интерфейса образовательной платформы; основные типы данных языка Python; правила использования переменных и базовых операций (арифметических и логических); синтаксис условных конструкций (if, elif, else); правила использования цикла while и операторов break, continue; возможности использования технологий искусственного интеллекта в программировании. уметь:	Введение в Python	Знакомство с образовательной платформой и особенностями обучения по программе. История создания языка Python. Особенности синтаксиса Python. Ввод и вывод данных. Основные типы данных языка Python: int, float, str, bool. Переменные и преобразование типов данных. Арифметические операции. Операции над строками. Форматирование строк. Логический тип данных. Поиск подстроки в строке.	Теоретическая подготовка	1
		Решение задач на ввод и вывод данных. Решение задач на использование переменных и типов данных. Решение задач на арифметические вычисления. Решение задач на обработку строк. Решение задач на поиск подстроки в строке. Решение задач на использование логических выражений. Разработка программ по заданному условию. Поиск и исправление ошибок в программном коде.	Практическая работа	10
		Решение задач на ввод и вывод данных. Решение задач на использование переменных и типов данных. Решение задач на арифметические вычисления. Решение задач на обработку строк. Решение задач на поиск подстроки в строке. Решение задач на использование логических выражений. Разработка программ по заданному условию. Анализ программного кода. Поиск и исправление ошибок в программном коде. Выполнение тестовых заданий на понимание основных понятий языка Python. Выполнение тестовых заданий на определение типов данных, результатов выражений и операций. Выполнение тестовых заданий на сопоставление конструкций языка Python и их назначения. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	5
		Решение задач на ввод и вывод данных. Решение задач на использование переменных и типов данных. Решение задач на арифметические вычисления. Решение задач на обработку строк. Решение задач на поиск подстроки в строке. Решение задач на использование логических выражений. Разработка программ по заданному условию.	Текущий контроль	1
	Условный оператор		Теоретическая подготовка	0

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

создавать простые программы на языке Python; использовать переменные и основные операции при решении задач; применять условные конструкции (if, elif, else); использовать цикл while для решения задач; применять операторы break и continue в циклах; использовать технологии искусственного интеллекта для анализа и улучшения программного кода; выполнять тестирование и отладку программ. владеть: навыками решения базовых алгоритмических задач; навыками работы в среде программирования; навыками разработки простых программ; навыками применения условий и циклов при решении задач; навыками проектной деятельности при выполнении индивидуального проекта.		Виды алгоритмов. Синтаксис условного оператора. Формы ветвления. Вложенный условный оператор. Каскадный условный оператор. Решение задач на выбор действия с использованием if, elif, else. Составление и проверка логических условий с операторами сравнения, and, or, not. Решение задач на проверку диапазонов и принадлежности (в т.ч. составные условия). Решение задач на обработку ввода с проверкой корректности данных и формата. Поиск и исправление ошибок в программах с ветвлением (логические и синтаксические ошибки).	Практическая работа	4
		Решение задач с условным оператором. Решение задач с вложенным оператором. Написание простых программ с условиями. Изучение дополнительных материалов по теме. Выполнение тестовых заданий на понимание работы с вложенными и каскадными условиями. Подготовка к текущему контролю по теме.	Самостоятельная работа	2
		Решение учебных задач с условным оператором if, else, elif. Поиск и исправление ошибок в программном коде.	Текущий контроль	1
	Использование ИИ-инструментов		Теоретическая подготовка	0
		ИИ-инструменты для программирования. Возможности и ограничения ИИ. Использование ИИ-инструментов в работе программиста. Разработка программ с использованием ИИ-инструмента. Поиск и исправление ошибок в коде, сгенерированном ИИ.	Практическая работа	1
		Разработка программ с использованием ИИ-инструмента. Поиск и исправление ошибок в коде, сгенерированном ИИ. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	1
			Текущий контроль	0
	Цикл с условием	Цикл while. Счетчик. Подсчет суммы с помощью цикла while. Условия в циклах. Операторы break, continue.	Теоретическая подготовка	1
		Поиск и исправление ошибок в программном коде. Решение задач на организацию повторяющихся действий с использованием цикла while и счётчика. Решение задач на накопление суммы/произведения и подсчёт количества значений с помощью while. Решение задач на ввод данных до выполнения условия остановки (ввод до «стоп»-значения, проверка корректности ввода). Решение задач с условиями внутри цикла: поиск первого подходящего значения, обработка граничных случаев. Решение задач с применением операторов управления циклом break и continue, поиск и исправление ошибок в программах с while.	Практическая работа	3
		Поиск и исправление ошибок в программном коде. Написание программы с числом попыток. Решение задач на использование цикла в программах. Выполнение тестовых заданий на понимание работы циклов, знание определения счетчика. Выполнение тестовых заданий на знание операторов continue и break. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	1
		Решение задач с циклами while, операторами управления break и continue. Поиск и исправление ошибок в программном коде.	Текущий контроль	1
	Проект		Теоретическая подготовка	0

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Марцина Виктория Вадимовна

серийный номер 018a31f1e3a1eeb04e8127ee019aedad

срок действия 05.12.2025 - 05.03.2027

		Повторение операторов ветвления и цикла while. Повторение арифметических операций (+, -, *, /, //, %, **). Обработка некорректного ввода и исключительных ситуаций (деление на ноль, неверная операция). Использование цикла while для организации непрерывной работы программы до выполнения условия остановки. Разработка проекта «Калькулятор». Реализация обработки арифметических операций и проверки корректности ввода. Реализация завершения работы программы по команде «стоп». Тестирование программы на различных вариантах входных данных.	Практическая работа	2
		Доработка проекта «Калькулятор». Тестирование программы на различных наборах входных данных. Поиск и исправление ошибок в программном коде.	Самостоятельная работа	1
			Текущий контроль	0
	Промежуточная аттестация	Практические задания на исправление ошибок в готовом программном коде. Практические задания на написание программного кода по предложенному алгоритму и описанию задачи. Задания на определение результата работы программного кода. Задания открытого типа. Комплекты из 15 практических заданий по теме Не менее 50% заданий составляют практико-ориентированные задания на написание программного кода Не менее 20% заданий составляют задания открытого типа Встроенная автоматизированная система проверки кода	Промежуточная аттестация	1
Модуль 2 Обработка строк Модуль направлен на развитие у обучающихся навыков обработки текстовой информации средствами языка Python, освоение функций и повторного использования программного кода, изучение цикла for и расширение алгоритмического мышления через решение практических задач и выполнение командного проекта. В рамках модуля обучающиеся изучают срезы строк, основные методы работы со строками (find(), replace(), count(), upper(), lower(), strip(), split(), join() и другие), кодирование символов (ASCII), функции и их параметры, области видимости переменных, рекурсию, цикл for, а также выполняют командный проект «Регистрация команд». Задачи модуля: сформировать навыки обработки текстовой информации средствами языка Python; познакомить со срезами строк и основными методами строк (find(), replace(), count(), upper(), lower(), strip(), split(), join() и другие); сформировать представление о кодировании символов и работе с Unicode; познакомить с созданием и использованием функций, включая параметры и возвращаемые значения; сформировать представление о локальных и глобальных переменных; познакомить с рекурсивными алгоритмами; сформировать навыки использования цикла for для обработки строк и последовательностей; развить навыки декомпозиции задач и командной работы при выполнении проекта. Планируемые результаты обучения: По итогам освоения модуля обучающийся будет:	Строки	Подстрока. Срезы строк. Шаг извлечения. Методы строк. Классификация строк. Проверка регистра. Методы изменения регистра. Методы поиска и замены.	Теоретическая подготовка	1
		Решение задач на извлечение подстроки. Решение задач на классификацию и проверку регистра. Решение задач на изменение регистра. Решение задач на поиск и замену строк. Разработка программ на подсчет слов в строке. Решение задач на кодирование символов.	Практическая работа	8
		Решение задач на извлечение подстроки. Решение задач на классификацию и проверку регистра. Решение задач на изменение регистра. Решение задач на поиск и замену строк. Разработка программ на подсчет слов в строке. Выполнение тестовых заданий на знание методов строк. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	5
		Решение практических задач на обработку строк с использованием индексов и срезов. Применение базовых методов строк lower(), upper(), split(), find(), replace() при решении учебных задач. Проверка формата строковых данных и преобразование регистра.	Текущий контроль	1
		Поиск и исправление ошибок в программном коде, связанном с обработкой строк.		
	Функции	Определение и вызов функций в Python. Понятие локальных и глобальных переменных. Возврат значений. Рекурсия. Глубина рекурсии. Стек вызова	Теоретическая подготовка	1

Контур Крипто
владелец Жарцева Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 06.12.2025 - 05.03.2027

По итогам освоения модуля обучающийся будет:

знать:

основные способы обработки строк в языке Python;
правила использования срезов строк;
назначение методов строк: find(), replace(), count(), upper(), lower(), strip(), split(), join() и другие;
принципы кодирования символов (ASCII);
принципы создания функций;
особенности локальных и глобальных переменных;
принципы рекурсии;
правила использования цикла for.

уметь:

использовать срезы строк при решении задач;
применять методы строк (find(), replace(), count(), upper(), lower(), strip(), split(), join() и другие);
выполнять преобразование строк и поиск подстрок;
создавать функции с параметрами и возвращаемыми значениями;
использовать локальные и глобальные переменные;
применять рекурсию в простых задачах;
использовать цикл for для обработки строк и последовательностей;
разрабатывать и тестировать программные решения;
выполнять командный проект.

владеть:

навыками обработки текстовой информации в Python;
навыками использования функций и рекурсивных алгоритмов;
навыками применения методов строк для решения практических задач;
навыками работы с циклами и последовательностями;
навыками декомпозиции задач;
навыками командной и проектной деятельности.

	Решение задач на создание функций с параметрами и возвратом значения (return). Разработка функций для типовых вычислений и обработки данных (строки/числа/коллекции). Использование функций для декомпозиции программы и устранения повторяющегося кода. Решение задач на функции с параметрами по умолчанию и именованными аргументами. Поиск и исправление ошибок в программах с функциями (область видимости, неверные аргументы, возврат значения).	Практическая работа	4
	Поиск и исправление ошибок в программном коде. Разработка программ с использованием функций. Решение задач с оператором return. Решение задач с рекурсией. Выполнение тестовых заданий на знание функций. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	2
	Решение задач на создание и использование функций (определение, вызов, параметры, возврат значения). Решение задач на применение рекурсивных функций. Поиск и исправление ошибок в программном коде, связанном с использованием функций и рекурсии.	Текущий контроль	1
Цикл for		Теоретическая подготовка	0
	Цикл for. Методы split и join. Решение задач с циклом for. Решение задач с методом split и join. Решение задач на перебор диапазонов с range() (в т.ч. шаг, обратный порядок). Решение задач на перебор элементов последовательностей (строки, списки). Решение задач на подсчёт, накопление суммы/произведения, поиск минимума/максимума в цикле. Решение задач на формирование новых последовательностей по правилу (генерация списков/строк). Поиск и исправление ошибок в программах с циклом for (границы диапазона, неверные условия, лишние действия в цикле). Выполнение командного проекта «Регистрация команд».	Практическая работа	4
	Решение задач с циклом for. Решение задач с методом split и join. Выполнение тестовых заданий на знание методов и циклов. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	2
	Решение задач на использование цикла for для перебора символов строки. Решение задач на использование функции range() для генерации числовых последовательностей. Решение задач на проверку вхождения символа в строку (операторы in, not in). Решение задач на использование методов строк (split(), join(), isdigit(), lower(), upper()). Решение задач на обработку строк с условиями внутри цикла (if, else, break). Решение задач на преобразование строк (удаление символов, замена, фильтрация). Разработка программ по заданному условию (проверка формата данных, фильтрация строк). Исправление ошибок в готовом коде (синтаксические ошибки в цикле for). Определение результата работы программного кода; задания открытого типа на знание синтаксиса цикла for и методов строк.	Текущий контроль	1
Командный проект		Теоретическая подготовка	0

Контур Крипто
владелец Марщеня Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01be2191e04f2cb94e6f27ce010eedbd
срок действия 05.12.2025 - 05.03.2027

		Понятие проекта, этапы разработки проекта и распределение задач в команде. Планирование и распределение задач при выполнении проекта. Разработка проекта «Регистрация команды». Объединение результатов работы участников, тестирование и отладка программы. Подготовка защиты проекта. Изучение дополнительных материалов по теме.	Практическая работа	4
		Доработка проекта и устранение ошибок. Тестирование программы на различных наборах входных данных. Подготовка презентации и защиты проекта.	Самостоятельная работа	1
			Текущий контроль	0
	Промежуточная аттестация	Практические задания на исправление ошибок в готовом программном коде. Практические задания на написание программного кода по предложенному алгоритму и описанию задачи. Задания на определение результата работы программного кода. Задания открытого типа. Комплекты из 15 практических заданий по теме Не менее 50% заданий составляют практико-ориентированные задания на написание программного кода Не менее 20% заданий составляют задания открытого типа Встроенная автоматизированная система проверки кода	Промежуточная аттестация	1
Модуль 3 Коллекции Модуль направлен на развитие у обучающихся навыков работы с коллекциями данных в языке Python, освоение структур данных для хранения и обработки информации, развитие алгоритмического мышления и навыков решения практических задач. В рамках модуля обучающиеся изучают списки, кортежи, основные методы списков (append(), remove(), pop(), insert()), сортировку данных, лямбда-функции, словари и их методы (keys(), values(), items(), get()), множества и операции над ними, а также выполняют проект «Книжный бот» и знакомятся с принципами ответственного использования технологий искусственного интеллекта. Задачи модуля сформировать навыки работы с коллекциями данных в языке Python; познакомить со списками и кортежами, их назначением и различиями; изучить основные методы списков (append(), remove(), pop(), insert()); сформировать навыки сортировки списков и использования лямбда-функций; познакомить со словарями и основными методами работы с ними; познакомить с множествами и операциями над ними; сформировать представление об ответственном использовании технологий искусственного интеллекта; развить навыки решения задач с использованием коллекций данных; развить навыки проектной деятельности при выполнении проекта. Планируемые результаты обучения	Списки и кортежи	Определение и создание списков и кортежей. Отличия списков и кортежей. Применение списков и кортежей в решении задач.	Теоретическая подготовка	1
		Решение задач на создание списков/кортежей и доступ к элементам по индексу и срезу. Решение задач на изменение списков (присваивание по индексу, добавление/удаление элементов). Решение задач на применение основных методов списков (append, insert, remove, pop, count, index). Решение задач на обработку списков в циклах (поиск, фильтрация, подсчёт, суммирование). Поиск и исправление ошибок в программах со списками и кортежами (индексация, копирование, изменение данных).	Практическая работа	5
		Выполнение операций со списками. Использование встроенных функций со списками при написании программ. Решение задач с использованием методов списков. Выполнение тестовых заданий на знание списков, кортежей. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	3
		Решение задач с использованием условного оператора if-elif-else и составных логических условий. Решение задач с циклами while и for, включая применение оператора continue. Выполнение практических заданий на базовые операции со списками: доступ по индексу, изменение элементов, обработка списка в цикле. Решение задач на преобразование строки в список с помощью split() и последующую обработку данных, поиск и исправление ошибок в коде.	Текущий контроль	1
	Ответственное использование ИИ-инструментов		Теоретическая подготовка	0
		Безопасность при использовании ИИ. Авторство и этика использования ИИ. Анализ решений, полученных от ИИ. Решение задач с использованием ИИ. Проверка кода, сгенерированного ИИ.	Практическая работа	1

Контур Крипто
владелец Марцюня Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

По итогам освоения модуля обучающийся будет: знать: назначение и особенности списков и кортежей; основные методы списков (append(), remove(), pop(), insert()); принципы сортировки данных в списках; основы использования лямбда-функций; структуру словарей и основные методы (keys(), values(), items(), get()); назначение множеств и операции над ними; принципы ответственного использования технологий искусственного интеллекта. уметь: создавать и использовать списки и кортежи при решении задач; применять методы списков (append(), remove(), pop(), insert()); сортировать списки и использовать лямбда-функции; работать со словарями и использовать их методы; выполнять операции над множествами; решать задачи с использованием различных коллекций данных; разрабатывать и тестировать программные решения; выполнять командный проект. владеть: навыками работы со списками, кортежами, словарями и множествами; навыками использования методов коллекций для обработки данных; навыками сортировки и структурирования данных; навыками решения алгоритмических задач с использованием коллекций; навыками проектной и командной деятельности при выполнении проекта; навыками ответственного использования технологий искусственного интеллекта.		Решение задач с использованием ИИ. Проверка кода, сгенерированного ИИ. Выполнение тестовых заданий на использование ИИ. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	1
			Текущий контроль	0
	Сортировка списков	Функции для сортировки. Лямбда-функции.	Теоретическая подготовка	1
		Решение задач со списками и кортежами. Написание функции для сортировки. Написание и использование лямбда-функции. Решение стандартных и нестандартных задач по программированию.	Практическая работа	4
		Решение задач со списками и кортежами. Написание функции для сортировки. Написание и использование лямбда-функции. Решение стандартных и нестандартных задач по программированию. Выполнение тестовых заданий на знание функций. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	2
		Решение задач на работу со списками: доступ к элементам по индексу, суммирование элементов, применение методов count() и remove(). Решение задач на преобразование строк в списки с использованием метода split() и последующую обработку полученных данных. Решение задач на сортировку списков с ключом (sorted(), key=len), а также применение filter() и lambda-выражений для отбора элементов. Поиск и исправление ошибок в программном коде, связанном с обработкой списков и сортировкой.	Текущий контроль	1
	Словари и множества		Теоретическая подготовка	0
		Знакомство со словарями. Методы для работы со словарями. Особенности множеств. Создание множеств. Функции и методы работы с множествами. Операции над множествами. Решение задач на создание словарей, добавление/изменение элементов и доступ к значениям по ключу. Решение задач на применение методов словарей (get(), keys(), values(), items(), update(), pop()) и перебор словаря. Решение задач на создание множеств и использование их свойства уникальности (удаление повторов, проверка принадлежности). Решение задач на применение методов множеств (add(), remove()/discard(), pop(), clear()) и преобразование между типами (список–множество). Решение задач на операции над множествами (объединение, пересечение, разность, симметрическая разность) и анализ результатов.	Практическая работа	8
		Решение задач с использованием словарей. Решение задач с использованием математических операций. Выполнение тестовых заданий на знание словарей, понимание работы со множествами. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	3
		Решение задач на работу со словарями: создание словаря, проверка наличия ключа, получение значения, вывод сообщения при отсутствии данных. Решение задач на обновление и пополнение словаря в цикле до стоп-слова с выводом итогового словаря. Решение задач на работу с множествами: получение уникальных символов строки и подсчёт их количества. Решение задач на операции над множествами: нахождение пересечения двух наборов чисел и вывод результата в отсортированном виде. Решение задач на проверку уникальности символов строки с использованием множеств и условного оператора.	Текущий контроль	1

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Марцюня Виктория Вадимовна

серийный номер 01ba319761441eeb04e6b27e6b19aeba0

срок действия 05.12.2025 - 05.03.2027

	Проект		Теоретическая подготовка	0
		Понятие проекта и этапы работы над программным продуктом. Повторение словарей, списков, циклов, ветвлений и функций. Анализ требований к проекту «Книжный-бот». Разработка программы с меню действий. Реализация вывода, поиска и добавления книг в базу. Тестирование, отладка и доработка проекта. Подготовка защиты проекта.	Практическая работа	2
		Доработка проекта «Книжный-бот». Проверка работы программы на разных вариантах входных данных. Подготовка демонстрации и защиты проекта.	Самостоятельная работа	1
			Текущий контроль	0
	Промежуточная аттестация	Практические задания на исправление ошибок в готовом программном коде. Практические задания на написание программного кода по предложенному алгоритму и описанию задачи. Задания на определение результата работы программного кода. Задания открытого типа. Комплекты из 15 практических заданий по теме Не менее 50% заданий составляют практико-ориентированные задания на написание программного кода Не менее 20% заданий составляют задания открытого типа Встроенная автоматизированная система проверки кода	Промежуточная аттестация	1
Модуль 4 Модуль Turtle Модуль направлен на развитие у обучающихся практических навыков программирования с использованием модулей Python, освоение графического программирования с использованием библиотеки turtle, знакомство с основами объектно-ориентированного программирования, развитие алгоритмического мышления и навыков создания интерактивных программ. В рамках модуля обучающиеся изучают подключение модулей (import), работу с генерацией случайных чисел (random), основы работы с библиотекой turtle, построение графических объектов, обработку событий, основы классов и объектов, принципы наследования, а также решают кейс-задачу «Пароль, который можно запомнить». Задачи модуля: сформировать навыки использования модулей Python (import, random); познакомить с генерацией случайных чисел и их использованием в программах; сформировать навыки работы с библиотекой turtle для создания графических программ; обучить построению графических фигур и простых интерактивных приложений; познакомить с базовыми принципами объектно-ориентированного программирования; сформировать представление о классах и объектах, событиях и наследовании; развить навыки создания игровых и графических приложений; развить навыки проектной деятельности при выполнении проекта; познакомить с решением прикладной кейс-задачи.	Модули в Python	Понятие модуля и импорт в Python. Импорт стандартных модулей: random, time.	Теоретическая подготовка	1
		Решение задач со случайными числами. .	Практическая работа	2
		Решение задач со случайными числами. Выполнение тестовых заданий на знание модулей. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	2
		Решение задач на использование модуля time: создание задержек, получение текущего времени, форматирование времени. Выполнение заданий на измерение времени выполнения фрагментов программы с использованием функций модуля time. Решение задач на генерацию случайных чисел с модулем random и управление воспроизводимостью результатов. Решение задач на выбор случайного элемента, перемешивание списка и моделирование простых случайных событий.	Текущий контроль	1
	Модуль Turtle	Модуль Turtle. Базовые команды модуля. Циклы и функции для отрисовки фигур и узоров. Способы оптимизации программ для рисования. Объектно-ориентированное программирование. События. Наследование.	Теоретическая подготовка	2
		Использование циклов в графических проектах с turtle. Выполнение проектов «Черепашьи гонки» и «Примитивный Paint». Выполнение практических заданий по написанию кода. Создание классов.	Практическая работа	13
		Использование циклов в графических проектах с turtle. Выполнение практических заданий по написанию кода. Создание классов. Выполнение тестовых заданий на знание модуля Turtle. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	6

Контур Крипто
владелец Маршеня Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Планируемые результаты обучения По итогам освоения модуля обучающийся будет: знать: назначение модулей Python и правила их подключения (import); назначение модуля random и генерации случайных чисел; возможности библиотеки turtle; основы объектно-ориентированного программирования; понятия классов, объектов, событий и наследования. уметь: использовать модули Python (import, random); применять генерацию случайных чисел в программах; создавать графические изображения с использованием turtle; строить простые интерактивные приложения; использовать основы классов и объектов при создании программ; разрабатывать и тестировать графические и игровые приложения; решать кейс-задачу. владеть: навыками работы с модулями Python; навыками создания графических программ с использованием turtle; навыками разработки интерактивных приложений; навыками базового объектно-ориентированного программирования; навыками проектной и командной деятельности; навыками решения прикладных задач и кейс-задач.		Решение задач на использование модулей time и random (задержки sleep(), получение текущего времени, генерация случайных чисел и выбор случайного элемента). Выполнение практических заданий в turtle: рисование базовых фигур (квадрат), повороты, настройка пера и заливка. Решение задач на основы ООП: создание класса, конструктор __init__, методы объекта, создание экземпляров. Поиск и исправление ошибок в программном коде, связанном с импортом модулей, использованием turtle и созданием классов.	Текущий контроль	1
	Кейс-задача		Теоретическая подготовка	0
		Анализ проблемной ситуации и формулирование требований к решению: какие пароли считаются надёжными и какие должны быть удобными для пользователя. Планирование командной разработки: распределение ролей, проектирование сценария работы продукта и критериев оценки пароля. Разработка прототипа или MVP решения на Python: проверка пароля по правилам, выдача рекомендаций по улучшению, генерация запоминаемых паролей или парольных фраз. Объединение результатов работы участников, тестирование и отладка программы на типовых и граничных случаях. Подготовка защиты кейс-задачи: демонстрация решения, объяснение принципов работы, анализ сильных сторон, ограничений и направлений развития, описание вклада участников.	Практическая работа	5
		Доработка функциональности и улучшение удобства использования решения. Подготовка набора тестов и проверка работы программы на разных сценариях, включая слабые и сильные пароли. Подготовка материалов для защиты: презентация, описание логики и критериев, предложения по развитию продукта.	Самостоятельная работа	2
			Текущий контроль	0
	Промежуточная аттестация	Практические задания на исправление ошибок в готовом программном коде. Практические задания на написание программного кода по предложенному алгоритму и описанию задачи. Задания на определение результата работы программного кода. Задания открытого типа. Комплекты из 15 практических заданий по теме Не менее 50% заданий составляют практико-ориентированные задания на написание программного кода Не менее 20% заданий составляют задания открытого типа Встроенная автоматизированная система проверки кода	Промежуточная аттестация	1
Итоговый контроль/аттестация	Итоговый контроль/аттестация по ДОП	Защита итогового проекта, разработанного в рамках выполнения кейс-задачи, включает демонстрацию полученного результата и ответы на вопросы согласно утвержденному плану: Обоснование выбранной проблемы. Цель проекта и ожидаемый результат. Предложенное решение. Распределение задач между участниками. Демонстрация результата и его работоспособности. Анализ сильных сторон, недостатков и ограничений решения. Перспективы развития проекта. Экспертный лист оценки проекта по установленным критериям Материалы итоговой защиты проекта	Итоговый контроль/аттестация	2.00

Контур Крипто
владелец Маршеня Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

ИТОГО ПО ПРОГРАММЕ:	Объем в ак.ч.	Объем в %
Теоретическая подготовка	9,00	6,16
Практическая работа	80,00	54,79
Самостоятельная работа	40,00	27,40
Текущий контроль	11,00	7,53
Промежуточная аттестация	4,00	2,74

	Итоговый контроль/аттестация	2.00	1,37
	Всего, ак.ч.:		**146**

ИТиСИ	Объем часов ИТиСИ не входит в общую трудоемкость ДОП, но участие в ИТиСИ обязательно для всех успешно прошедших Итоговый контроль/аттестацию по ДОП

Календарный учебный график

№ п/п	Название ДОП	№ потока	Дата начала обучения по ДОП	№ модуля ДОП	Начало обучения по модулю ДОП	Календарный период (количество дней) длительности модуля	Дата окончания обучения по ДОП
1	Python Старт (базовый уровень, 8–9 классы)	1	2026-09-28	1	2026-09-28	64	2027-05-31
				2	2026-11-16	79	
				3	2027-01-11	79	
				4	2027-03-01	92	

Сведения об условиях реализации ДОП (в том числе материально-техническое, информационное, кадровое обеспечение)

Материально-технические и информационные условия реализации ДОП

Наименование поля	Значение полей	Значение полей	Значение полей	Значение полей
Порядковый номер модуля	1	2	3	4
Перечень требований к оснащению площадки (для офлайн-формата)	Не заполняется	Не заполняется	Не заполняется	Не заполняется
Наименование требуемого оборудования (для онлайн-формата)	Компьютер или ноутбук с доступом к сети Интернет. Процессор: не менее 2 физических ядер с тактовой частотой от 2,0 ГГц. Оперативная память: не менее 4 ГБ (рекомендуется 8 ГБ и более). Свободное место на накопителе: не менее 4 ГБ. Разрешение экрана: не менее 1366×768 пикселей. Устройства ввода: клавиатура и мышь или иные устройства, обеспечивающие полноценную работу с компьютером. Наушники или колонки. Микрофон. Веб-камера (рекомендуется для участия в онлайн-занятиях).	Компьютер или ноутбук с доступом к сети Интернет. Процессор: не менее 2 физических ядер с тактовой частотой от 2,0 ГГц. Оперативная память: не менее 4 ГБ (рекомендуется 8 ГБ и более). Свободное место на накопителе: не менее 4 ГБ. Разрешение экрана: не менее 1366×768 пикселей. Устройства ввода: клавиатура и мышь или иные устройства, обеспечивающие полноценную работу с компьютером. Наушники или колонки. Микрофон. Веб-камера (рекомендуется для участия в онлайн-занятиях).	Компьютер или ноутбук с доступом к сети Интернет. Процессор: не менее 2 физических ядер с тактовой частотой от 2,0 ГГц. Оперативная память: не менее 4 ГБ (рекомендуется 8 ГБ и более). Свободное место на накопителе: не менее 4 ГБ. Разрешение экрана: не менее 1366×768 пикселей. Устройства ввода: клавиатура и мышь или иные устройства, обеспечивающие полноценную работу с компьютером. Наушники или колонки. Микрофон. Веб-камера (рекомендуется для участия в онлайн-занятиях).	Компьютер или ноутбук с доступом к сети Интернет. Процессор: не менее 2 физических ядер с тактовой частотой от 2,0 ГГц. Оперативная память: не менее 4 ГБ (рекомендуется 8 ГБ и более). Свободное место на накопителе: не менее 4 ГБ. Разрешение экрана: не менее 1366×768 пикселей. Устройства ввода: клавиатура и мышь или иные устройства, обеспечивающие полноценную работу с компьютером. Наушники или колонки. Микрофон. Веб-камера (рекомендуется для участия в онлайн-занятиях).

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Наименование требуемого программного обеспечения (для онлайн-формата)	Операционная система: Windows 10 и выше, Linux или macOS. Современный веб-браузер с поддержкой актуальных веб-стандартов (рекомендуются последние версии Google Chrome или Яндекс Браузера). Среда разработки Python Wing 101 версии 10.0.8 и выше или другая совместимая среда разработки Python. Интерпретатор Python версии 3.10 и выше (при необходимости выполнения практических заданий на локальном компьютере). Для прохождения обучения достаточно использования образовательной платформы и современного веб-браузера. Установка интерпретатора Python и среды разработки рекомендуется для выполнения практических заданий на локальном компьютере и не является обязательной.	Операционная система: Windows 10 и выше, Linux или macOS. Современный веб-браузер с поддержкой актуальных веб-стандартов (рекомендуются последние версии Google Chrome или Яндекс Браузера). Среда разработки Python Wing 101 версии 10.0.8 и выше или другая совместимая среда разработки Python. Интерпретатор Python версии 3.10 и выше (при необходимости выполнения практических заданий на локальном компьютере). Для прохождения обучения достаточно использования образовательной платформы и современного веб-браузера. Установка интерпретатора Python и среды разработки рекомендуется для выполнения практических заданий на локальном компьютере и не является обязательной.	Операционная система: Windows 10 и выше, Linux или macOS. Современный веб-браузер с поддержкой актуальных веб-стандартов (рекомендуются последние версии Google Chrome или Яндекс Браузера). Среда разработки Python Wing 101 версии 10.0.8 и выше или другая совместимая среда разработки Python. Интерпретатор Python версии 3.10 и выше (при необходимости выполнения практических заданий на локальном компьютере). Для прохождения обучения достаточно использования образовательной платформы и современного веб-браузера. Установка интерпретатора Python и среды разработки рекомендуется для выполнения практических заданий на локальном компьютере и не является обязательной.	Операционная система: Windows 10 и выше, Linux или macOS. Современный веб-браузер с поддержкой актуальных веб-стандартов (рекомендуются последние версии Google Chrome или Яндекс Браузера). Среда разработки Python Wing 101 версии 10.0.8 и выше или другая совместимая среда разработки Python. Интерпретатор Python версии 3.10 и выше (при необходимости выполнения практических заданий на локальном компьютере). Для прохождения обучения достаточно использования образовательной платформы и современного веб-браузера. Установка интерпретатора Python и среды разработки рекомендуется для выполнения практических заданий на локальном компьютере и не является обязательной.

Информационные условия реализации ДОП (заполняется по каждому модулю):

Наименование поля	Значение полей	Значение полей	Значение полей	Значение полей
Порядковый номер модуля	1	2	3	4
Электронные информационные ресурсы	При реализации программы могут использоваться следующие электронные информационные и образовательные ресурсы: https://docs.python.org/3/ — официальная документация Python; https://proproprogs.ru/python — справочные материалы по Python; https://pythontutor.com/ — визуализация выполнения программ	При реализации программы могут использоваться следующие электронные информационные и образовательные ресурсы: https://docs.python.org/3/ — официальная документация Python; https://proproprogs.ru/python — справочные материалы по Python; https://pythontutor.com/ — визуализация выполнения программ	При реализации программы могут использоваться следующие электронные информационные и образовательные ресурсы: https://docs.python.org/3/ — официальная документация Python; https://proproprogs.ru/python — справочные материалы по Python; https://pythontutor.com/ — визуализация выполнения программ	При реализации программы могут использоваться следующие электронные информационные и образовательные ресурсы: https://docs.python.org/3/ — официальная документация Python; https://proproprogs.ru/python — справочные материалы по Python; https://pythontutor.com/ — визуализация выполнения программ https://docs.python.org/3/library/turtle.html — документация по Turtle;

Наименование поля	Значение полей	Значение полей	Значение полей	Значение полей
Порядковый номер модуля	1	2	3	4
Электронные образовательные ресурсы	При реализации программы могут использоваться следующие электронные информационные и образовательные ресурсы: https://pythontutor.ru/ — задачи по основам Python; https://informatics.msk.ru/ — задачи по алгоритмизации; https://acmp.ru/ — задачи начального уровня	При реализации программы могут использоваться следующие электронные информационные и образовательные ресурсы: https://pythontutor.ru/ — задачи по основам Python; https://informatics.msk.ru/ — задачи по алгоритмизации; https://acmp.ru/ — задачи начального уровня	При реализации программы могут использоваться следующие электронные информационные и образовательные ресурсы: https://pythontutor.ru/ — задачи по основам Python; https://informatics.msk.ru/ — задачи по алгоритмизации; https://acmp.ru/ — задачи начального уровня	При реализации программы могут использоваться следующие электронные информационные и образовательные ресурсы: https://pythontutor.ru/ — практические задачи по программированию; https://informatics.msk.ru/ — проектные задачи; https://stepindev.com/ru/py-playground — выполнение и публикация проектов

Кадровое обеспечение

Сведения о работниках, привлекаемых к реализации и разработке ДОП

Контур Крипто
владелец Маршеня Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

ФИО	Участие в реализации ДОП	Наименование основного места работы	Ученая степень и (или) ученое звание (при наличии)	Образование (наименование учебного заведения, полученная специальность, год окончания)	Основание для привлечения к реализации ДОП (трудовой договор, договор ГПХ, иное)	Информация о дополнительном профессиональном образовании по профилю преподаваемой дисциплины (за последние 3 года)	Опыт работы в области ИТ, ИИ или робототехники и (или) профессиональные достижения в области ИТ, ИИ или робототехники (награды, научные статьи, иное)	Отметка о полученном согласии на обработку персональных данных
строка	(разработка/ реализация)	строка	строка	строка	строка	строка	строка	да/ нет
Подъельников Лев Андреевич	Разработка	Учи.Дома Ведущий методист		Чувашский государственный педагогический университет имени И.Я. Яковлева 2010 Физико-математический, Учитель математики и информатики	Трудовой договор			Да
Филиппов Егор Игоревич	Реализация	Учи.Дома Преподаватель		Федеральное государственное автономное образовательное учреждение высшего образования “Самарский национальный исследовательский университет имени академика С.П. Королева” 2022 Бизнес-информатика	Иное: Самозанятость	Федеральное государственное автономное образовательное учреждение высшего образования “Самарский национальный исследовательский университет имени академика С.П. Королева”, Системы искусственного интеллекта, Системы компьютерного зрения, Python для анализа данных, 2022г		Да
Лютов Егор Вячеславович	Реализация	Учи.Дома Преподаватель		Федеральное государственное бюджетное образовательное учреждение высшего образования "Уфимский государственный авиационный технический университет" 2026 Авионика, энергетика и инфокоммуникации	Иное: Самозанятость			Да
Гнездилова Ангелина Дмитриевна	Реализация	Учи.Дома Преподаватель		Федеральное государственное бюджетное образовательное учреждение высшего образования “Юго-Западный государственный университет” 2022 Математическое обеспечение и администрирование информационных систем	Иное: Самозанятость			Да
Толпеев Иван Васильевич	Реализация	Учи.Дома Преподаватель		Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования "Краснодарский государственный университет культуры и искусств" 2015 Менеджер социально-культурной деятельности	Иное: ИП	Общество с ограниченной ответственностью "Учи.ру", Особенности обучения программированию на курсе Python Start + AI в рамках проекта "Код Будущего", 2024	Разработчик, обл. ИИ, сайты, база данных, crm, 3 года.	Да
Афанасьев Иван Александрович	Реализация	Учи.Дома Преподаватель		Федеральное государственное автономное образовательное учреждение высшего профессионального образования "Сибирский федеральный университет" 2012 Машины и оборудование нефтяных и газовых промыслов	Иное: Самозанятость	"Институт дополнительного образования и повышения квалификации, Обучение английскому языку в общеобразовательных организациях по направлению ""образование и педагогика"", 2020г АНО ДПО ""ШАД"", Python-разработчик, 2022г		Да
Асеев Ринат Раифович	Реализация	Учи.Дома Преподаватель		Уральский государственный педагогический университет 2013 Информационные технологии в образовании	Иное: Самозанятость		Инженер - программист, сети, программирование, 18 лет	Да
Небольсин Анатолий Александрович	Реализация	Учи.Дома Преподаватель		Федеральное государственно бюджетное образовательное учреждение высшего образования "Воронежский государственный университет инженерных технологий" 2024 Информационная безопасность автоматизированных систем	Иное: Самозанятость		нейронка от яндекс, тестировщик, около 3-4 месяцев	Да

Контур Крипто
владелец Марцева Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Семенников Владислав Дмитриевич	Реализация	Учи.Дома Преподаватель		Федеральное государственное бюджетное образовательное учреждение высшего образования “Саратовский национальный исследовательский государственный университет имени Н.Г. Чернышевского” 2022 Информационные системы и программирование	Иное: Самозанятость			Да
Голубь Борис Семенович	Реализация	Учи.Дома Преподаватель		Московский институт радиотехники, электроники и автоматики 1985 Конструирование и производство электронно-вычислительной аппаратуры	Иное: Самозанятость	Государственное бюджетное образовательное учреждение дополнительного профессионального образования Центр “Профессионал”, Программирование на языке Python, 2022г	Платы для ввода изображения в комп, схематехника, программное обеспечение, микроконтролер, лет 6 - небольшая компания своя была, умный дом для себя сделал. Технический директор в телевидении лет 7, 16 лет возглавлял отдел строительство телевидения - проектные менеджеры	Да
Шлыков Вадим Николаевич	Реализация	Учи.Дома Преподаватель		Негосударственное аккредитованное частное образовательное учреждение высшего образования "Академия маркетинга и социально-информационных технологий - ИМСИТ" 2016 Программирование в компьютерных системах	Иное: Самозанятость			Да
Комарова Екатерина Юрьевна	Реализация	Учи.Дома преподаватель		Государственное образовательное учреждение высшего профессионального образования “Московский государственный индустриальный университет” 2009 Математическое обеспечение и администрирование информационных систем	Иное: Самозанятость			Да
Рожков Арсений Владимирович	Реализация	Учи.Дома Преподаватель		Московский государственный технический университет им. Н. Э. Баумана 2022 Прикладная информатика	Иное: Самозанятость		6 мес, младший разработчик пайтон, инженер систем, обл. программирование, до этого с++ - разработка, схожая позиция 1 год.	Да

Формы аттестации и оценочные материалы (комплект контрольно-измерительных материалов, позволяющих определить достижение Получателями поддержки планируемых результатов обучения)

Наименование поля	Значение полей
Текущий контроль	
Модуль 1, Первая программа, Введение в Python	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи
Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполнено не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Маршеня Виктория Вадимовна
срок действия 05.12.2025 - 05.03.2027

Варианты заданий	Задача 1. Исправь ошибки в выводе чисел и текста
	Дана программа с ошибками:
	print(2024 512)
	print('result =', 15)
	print(Люблю программировать)
	Найди и исправь ошибки так, чтобы программа выводила три строки.
	Входные данные
	Ничего не вводится.
	Выходные данные
	Три строки:
	2024 512
	result = 15
	Люблю программировать
	Решение
	print(2024, 512) # запятая вместо пробела
	print('result =', 15)
	print('Люблю программировать') # строка в кавычках
	Задача 2. Переменная и изменение значения
	Создай переменную code и присвой ей значение 256. Выведи эту переменную. Затем измени значение переменной на строку «Мария» и выведи сообщение:
	Добро пожаловать, Мария.
	Входные данные
	Ничего не вводится.
	Выходные данные
	Две строки:
	256
	Добро пожаловать, Мария
	Решение
	code = 256
	print(code)
	code = 'Мария'
	print('Добро пожаловать,', code)
	Задача 3. Сумма двух дробных чисел
	Напиши программу, которая запрашивает два вещественных числа (каждое на отдельной строке) и выводит их сумму.
	Входные данные
	Два вещественных числа, каждое на отдельной строке.
	Выходные данные
	Одно вещественное число – сумма.
	Пример

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

	Ввод: 4.25 3.75 Вывод: 8.0 Примечание Для вывода используй только print(), подсказки внутри input() не используй. Решение a = float(input()) b = float(input()) print(a + b) Задача 4. Вычисление по формуле Запрашиваются два целых числа a и b. Вычисли значение выражения: 5 * a**2 + 3 * b**(3+1) Входные данные Два целых числа, каждое на отдельной строке – a и b. Выходные данные Одно целое число – результат вычисления. Пример Ввод: 2 3 Вывод: 263 Примечание Порядок действий: сначала возведение в степень, затем умножение, потом сложение. Подсказки внутри input() не используй. Решение a = int(input()) b = int(input()) print(5 * a**2 + 3 * b**(3 + 1)) Задача 5. Проверка категории и цены (магазин электроники) Магазин электроники: остались товары категории «ноутбуки» по цене от 25 000 до 120 000 рублей включительно. Программа запрашивает у пользователя категорию товара и максимальную цену, которую он готов заплатить. Выводит: Товар есть - True, если категория равна «ноутбуки» и цена находится в указанном диапазоне; Товар есть - False в противном случае. Входные данные
	Первая строка – категория (строка). Вторая строка – целое число (цена). Выходные данные Строка вида Товар есть - True или Товар есть - False. Примеры Ввод: ноутбуки 30000 Вывод: Товар есть - True Ввод: планшеты 50000 Вывод: Товар есть - False Примечание Подсказки внутри input() не используй. Решение category = input() price = int(input()) result = (category == 'ноутбуки') and (25000 <= price <= 120000) print('Товар есть -', result)
	Модуль 1, Первая программа, Условный оператор
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
Владелец Маршеня Виктория Вадимовна
серийный номер 01ba2197c94f2cb84c6f27ee919aedad
срок действия 05.12.2025 - 05.03.2027

Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполняет не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)

Варианты заданий	Задача 1. Исправь ошибки в условном операторе
	Дана программа с ошибками:
	print('Введите число:')
	a = int(input())
	if a > 0
	print('Число положительное')
	Else:
	print('Число не положительное')
	Найди и исправь ошибки так, чтобы программа корректно определяла положительное число.
	Входные данные
	Одно целое число.
	Выходные данные
	Одна строка.
	Пример
	Ввод:
	-5
	Вывод:
	Число не положительное
	Примечание
	Подсказки внутри input() не используй.
	Решение
	print('Введите число:')
	a = int(input())
	if a > 0:
	  print('Число положительное')
	else:
	  print('Число не положительное')
	Задача 2. Проверка победителя
	Напиши программу, которая запрашивает у пользователя количество набранных очков. Если количество очков больше или равно 15, программа выводит «Победа!», иначе – «Поражение».
	Входные данные
	Одно целое число.
	Выходные данные
	Строка.
	Пример
	Ввод:
	12
	Вывод:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Поражение
Примечание
Подсказки внутри input() не используй.
Решение

```
score = int(input())
if score >= 15:
    print('Победа!')
else:
    print('Поражение')
```

Задача 3. Стоимость билета со скидкой
Билет в музей для взрослых (возраст ≥ 18 лет) стоит 450 рублей, а для детей (возраст < 18 лет) предоставляется скидка 15%. Напиши программу, которая запрашивает возраст пользователя и определяет стоимость билета с учётом возможной скидки.
Входные данные
Одно целое число – возраст.
Выходные данные
Одно вещественное число – стоимость билета.
Пример
Ввод:

```
14
```

Вывод:

```
382.5
```

Примечание
Подсказки внутри input() не используй.
Решение

```
age = int(input())
price = 450.0
if age < 18:
    print(price * 0.85)
else:
    print(price)
```

Задача 4. Преобразование двузначного числа
Пользователь вводит двузначное целое число. Напиши программу, которая вычисляет сумму его цифр. Если сумма цифр чётная и меньше 8, то программа умножает исходное число на 3, в противном случае – вычитает из числа 5. Выведите полученное число.
Входные данные
Одно целое число (10–99).
Выходные данные
Одно целое число.
Примеры
Ввод:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

	24
	(сумма цифр 2+4=6 – чётная и <8 → умножаем на 3)
	Вывод:
	72
	Ввод:
	89
	(сумма цифр 8+9=17 – не подходит под условие → вычитаем 5)
	Вывод:
	84
	Примечание
	Подсказки внутри input() не используй.
	Решение
	num = int(input())
	digit1 = num // 10
	digit2 = num % 10
	sum_digits = digit1 + digit2
	if sum_digits % 2 == 0 and sum_digits < 8:
	print(num * 3)
	else:
	print(num - 5)
	Задача 5. Наибольшее из трёх чисел
	Напиши программу, которая запрашивает три целых числа и выводит наибольшее из них.
	Входные данные
	Три целых числа, каждое на отдельной строке.
	Выходные данные
	Одно целое число – максимум.
	Примеры
	Ввод:
	7
	2
	9
	Вывод:
	9
	Ввод:
	5
	5
	3

	Вывод:
	5
	Примечание
	Используй конструкцию if-elif-else. Подсказки внутри input() не используй.
	Решение
	a = int(input())
	b = int(input())
	c = int(input())
	if a >= b and a >= c:
	print(a)
	elif b >= a and b >= c:
	print(b)
	else:
	print(c)

Модуль 1, Первая программа, Использование ИИ-инструментов	
Количество академических часов	
Формы контроля	
Диагностические инструменты	
Показатели и критерии оценивания	
Модуль 1, Первая программа, Цикл с условием	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи
Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода

Контур Крипто
владелец Маршеня Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba2191e94f2eb84e6f27ee819aedad
срок действия с 15.12.2025 по 05.03.2027

Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполняет не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)
Варианты заданий	Задача 1. Исправь ошибки в программе с паролем Дана программа (счётчик попыток) с ошибками: count = 1 password = 'qwerty' print('Введите пароль:') answer = int(input()) while password != answer: print('Пароль неверен') print('Введите пароль:') answer = int(input()) count += 1 print(f'Пароль верен. Вы угадали с {count} попытки') Найди и исправь ошибки. Программа должна запрашивать пароль до тех пор, пока не будет введён верный пароль qwerty. При каждом неверном вводе выводить сообщение и увеличивать счётчик попыток. При верном пароле выводить сообщение с номером попытки. Входные данные Строки (пароли) до тех пор, пока не будет введён qwerty. Выходные данные Для каждого неверного пароля: Пароль неверен и повторный запрос. При верном пароле: Пароль верен. Вы угадали с N попытки. Пример работы Ввод: 123 йцу qwerty Вывод: Введите пароль: Пароль неверен Введите пароль: Пароль неверен Введите пароль: Пароль верен. Вы угадали с 3 попытки Примечание Пароль – строка, не число. Подсказки внутри input() не используй. Решение count = 1 password = 'qwerty' print('Введите пароль:')

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
answer = input()          # исправлено: input() вместо int(input())
while password != answer:
    print('Пароль неверен')
    print('Введите пароль:')
    answer = input()
    count += 1
print(f'Пароль верен. Вы угадали с {count} попытки')
```

Задача 2. Вывод чисел от 1 до N

Напиши программу, которая запрашивает целое число n и выводит в столбик все целые числа от 1 до n включительно.

Входные данные

Одно целое число.

Выходные данные

n чисел, каждое на отдельной строке.

Пример

Ввод:

```
5
```

Вывод:

```
1
2
3
4
5
```

Примечание

Подсказки внутри input() не используй.

Решение

```
n = int(input())
i = 1
while i <= n:
    print(i)
    i += 1
```

Задача 3. Сумма цифр числа

Напиши программу, которая запрашивает целое положительное число и вычисляет сумму его цифр.

Входные данные

Одно целое число.

Выходные данные

Одно целое число – сумма цифр.

Пример

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Ввод:
215
Вывод:
8
Примечание
Подсказки внутри input() не используй.
Решение

```
n = int(input())
total = 0
while n > 0:
    digit = n % 10
    total += digit
    n //= 10
print(total)
```

Задача 4. Сумма чисел до нуля
Напиши программу, которая запрашивает целые числа и суммирует их, пока не будет введён 0. При вводе 0 программа выводит полученную сумму и завершается.
Входные данные
Несколько целых чисел (каждое на отдельной строке). Последнее число – 0.
Выходные данные
Одно целое число – сумма всех введённых чисел (без учёта завершающего нуля).
Пример
Ввод:
1
2
3
0
Вывод:
6
Примечание
Подсказки внутри input() не используй.
Решение

```
x = int(input())
total = 0
while x != 0:
    total += x
    x = int(input())
print(total)
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

	Задача 5. Игра «Угадай число»
	Компьютер загадал число 78. Напиши программу, которая:
	запрашивает у пользователя целое число;
	если число меньше 78 – выводит «Загаданное число больше»;
	если число больше 78 – выводит «Загаданное число меньше»;
	если число равно 78 – выводит «Ты угадал с N попытки» (где N – номер попытки) и завершает программу;
	если пользователь вводит 220 – программа завершается без всякого вывода (игрок сдался).
	Входные данные
	Несколько целых чисел (каждое на отдельной строке), пока не будет введено 78 или 220.
	Выходные данные
	При неверных догадках – подсказки (Загаданное число больше / меньше).
	При угадывании – сообщение о победе.
	При вводе 220 – ничего не выводится.
	Пример работы (победа)
	Ввод:
	45
	95
	78
	Вывод:
	Загаданное число больше
	Загаданное число меньше
	Ты угадал с 3 попытки
	Пример работы (сдался)
	Ввод:
	100
	220
	Вывод:
	Загаданное число меньше
	Примечание
	Подсказки внутри input() не используй.
	Решение
	count = 1
	secret = 78
	a = int(input())
	while a != 220:
	  if a == secret:
	  print(f'Ты угадал с {count} попытки')

	break
	  elif a < secret:
	  print('Загаданное число больше')
	  else:
	  print('Загаданное число меньше')
	  a = int(input())
	  count += 1

Модуль 1, Первая программа, Проект	
Количество академических часов	
Формы контроля	
Диагностические инструменты	
Показатели и критерии оценивания	
Модуль 2, Обработка строк, Строки	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи
Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполняет не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Маринова Виктория Вадимовна
серийный номер 01ab5191c94fe2024e0127e50 заверен
срок действия 05.12.2025 - 05.03.2027

Варианты заданий	Задача 1. Средний балл из строки с оценками
	В электронном дневнике оценки хранятся в строке формата:
	«Оценки за четверть: 3, 4, 5» (три оценки, разделённые запятыми и пробелами).
	Напиши программу, которая извлекает из строки три оценки (каждую – как целое число), вычисляет их среднее арифметическое и выводит результат (вещественное число).
	Входные данные
	Одна строка в формате «Оценки за четверть: a, b, c», где a, b, c – целые числа от 2 до 5.
	Выходные данные
	Одно вещественное число – средняя оценка.
	Пример
	Ввод:
	Оценки за четверть: 4, 5, 3
	Вывод:
	4.0
	Примечание
	Положение цифр в строке всегда одинаковое. Подсказки внутри input() не используй.
	Решение
	s = input()
	# цифры находятся на позициях 18, 22, 26 (при нумерации с 0)
	a = int(s[18])
	b = int(s[22])
	c = int(s[26])
	print((a + b + c) / 3)
	Задача 2. Срез строки по индексам
	Напиши программу, которая запрашивает строку, затем два целых числа:
	start (включительно) и stop (не включительно).
	Программа выводит срез строки от start до stop.
	Входные данные
	Первая строка – произвольный текст.
	Вторая строка – целое число start.
	Третья строка – целое число stop.
	Выходные данные
	Одна строка – результат среза.
	Пример
	Ввод:
	Привет, мир!
	6
	10

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Вывод:
т, м
Примечание
Подсказки внутри input() не используй.
Решение
```
text = input()
start = int(input())
stop = int(input())
print(text[start:stop])
```
Задача 3. Переворот строки
Напиши программу, которая запрашивает у пользователя строку и выводит её «задом наперёд» (в обратном порядке символов).
Входные данные
Одна строка.
Выходные данные
Одна строка – перевёрнутая исходная.
Пример
Ввод:
Программирование
Вывод:
еинавориммаргорП
Примечание
Подсказки внутри input() не используй.
Решение
```
s = input()
print(s[::-1])
```
Задача 4. Проверка номера телефона
Правильный номер телефона должен:
начинаться с цифры 8;
содержать ровно 11 цифр (все символы – цифры).
Напиши программу, которая проверяет введённую строку на соответствие этому формату.
Если номер верный – вывести «Номер правильный», иначе – «Номер неправильный».
Входные данные
Одна строка (телефонный номер).
Выходные данные
Строка «Номер правильный» или «Номер неправильный».
Примеры
Ввод:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

	89271234567
	Вывод:
	Номер правильный
	Ввод:
	+79271234567
	Вывод:
	Номер неправильный
	Примечание
	Подсказки внутри input() не используй.
	Решение
	phone = input()
	if phone[0] == '8' and len(phone) == 11 and phone.isdigit():
	  print('Номер правильный')
	else:
	  print('Номер неправильный')
	Задача 5. Проверка логина
	Логин пользователя считается корректным, если:
	он состоит только из букв (латиница или кириллица) и цифр (без пробелов и спецсимволов);
	не начинается с цифры.
	Напиши программу, которая проверяет введённую строку на эти условия и выводит True (если логин корректен) или False.
	Входные данные
	Одна строка – логин.
	Выходные данные
	True или False.
	Примеры
	Ввод:
	superman777
	Вывод:
	True
	Ввод:
	7superman
	Вывод:
	False
	Ввод:
	user name
	Вывод:
	False

	Примечание Подсказки внутри input() не используй. Решение login = input() print(login.isalnum() and not login[0].isdigit())
Модуль 2, Обработка строк, Функции	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи
Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполняет не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Варианты заданий	Задача 1. Исправь ошибки в функции (параметры)
	Дана программа с ошибками:
	def count_cats()
	  print("Всего котиков:", a + b)
	
	count_cats(5, 7)
	Найди и исправь ошибки так, чтобы программа выводила Всего котиков: 12.
	Входные данные
	Ничего не вводится.
	Выходные данные
	Одна строка.
	Пример вывода
	Всего котиков: 12
	Решение
	def count_cats(a, b):
	  print("Всего котиков:", a + b)
	
	count_cats(5, 7)
	Задача 2. Функция приветствия с параметром
	Напиши программу, в которой создай функцию greet(name). Функция должна выводить строку в формате:
	«Добрый вечер, name! Рады вас видеть.».
	Вызови функцию три раза с именами: «Анна», «Марк», «Лена».
	Входные данные
	Ничего не вводится.
	Выходные данные
	Три строки.
	Пример вывода
	Добрый вечер, Анна! Рады вас видеть.
	Добрый вечер, Марк! Рады вас видеть.
	Добрый вечер, Лена! Рады вас видеть.
	Решение
	def greet(name):
	  print(f'Добрый вечер, {name}! Рады вас видеть.')
	
	greet('Анна')
	greet('Марк')
	greet('Лена')

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Задача 3. Функция «Время года» (с return)
Напиши функцию season(month), которая получает номер месяца (целое число) и возвращает строку:
для месяцев 12, 1, 2 – «Зима»;
для 3, 4, 5 – «Весна»;
для 6, 7, 8 – «Лето»;
для 9, 10, 11 – «Осень»;
для остальных значений – «Нет месяца с таким номером».
В основной программе запроси номер месяца, вызови функцию и выведи результат.
Входные данные
Одно целое число – номер месяца.
Выходные данные
Одна строка.
Примеры
Ввод:
5
Вывод:
Весна
Ввод:
13
Вывод:
Нет месяца с таким номером
Примечание
Подсказки внутри input() не используй.
Решение

```
def season(month):
    if month == 12 or month == 1 or month == 2:
        return 'Зима'
    elif month == 3 or month == 4 or month == 5:
        return 'Весна'
    elif month == 6 or month == 7 or month == 8:
        return 'Лето'
    elif month == 9 or month == 10 or month == 11:
        return 'Осень'
    else:
        return 'Нет месяца с таким номером'

month = int(input())
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

	print(season(month))
	Задача 4. Рекурсивная сумма чисел от 1 до n
	Напиши рекурсивную функцию sum_to(n), которая возвращает сумму всех целых чисел от 1 до n включительно.
	Используй рекурсию (не циклы).
	В основной программе запроси число n и выведи результат.
	Входные данные
	Одно целое число n (n ≥ 1).
	Выходные данные
	Одно целое число – сумма.
	Пример
	Ввод:
	5
	Вывод:
	15
	Примечание
	Подсказки внутри input() не используй.
	Решение
	def sum_to(n):
	  if n == 1:
	  return 1
	  return n + sum_to(n - 1)
	
	n = int(input())
	print(sum_to(n))
	Задача 5. Рекурсивное возведение в степень (без **)
	Напиши рекурсивную функцию power(n, p), которая вычисляет n в степени p (без использования оператора ** или pow).
	Считай, что p – натуральное число (≥ 1).
	В основной программе запроси два целых числа n и p и выведи результат.
	Входные данные
	Два целых числа, каждое на отдельной строке: сначала n, затем p.
	Выходные данные
	Одно целое число.
	Пример
	Ввод:
	2
	5
	Вывод:

	32
	Примечание
	Подсказки внутри input() не используй.
	Решение
	def power(n, p):
	  if p == 1:
	  return n
	  return n * power(n, p - 1)
	
	n = int(input())
	p = int(input())
	print(power(n, p))

Модуль 2, Обработка строк, Цикл for	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи
Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполняет не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Мершина Виктория Калимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Варианты заданий	Задача 1. Исправь ошибки в цикле for для строки
	Дана программа с ошибками:
	letter = 'письмо'
	For in letter
	  print(i)
	Найди и исправь ошибки. Программа должна выводить каждый символ строки на отдельной строке.
	Входные данные
	Ничего не вводится.
	Выходные данные
	Символы строки на отдельных строках.
	Пример вывода
	п
	и
	с
	ь
	м
	о
	Решение
	letter = 'письмо'
	for i in letter:
	  print(i)
	Задача 2. Таблица умножения на 3
	Напиши программу, которая выводит таблицу умножения на 3 в столбик (от 1 до 9) в формате:
	3 * 1 = 3
	3 * 2 = 6
	...
	3 * 9 = 27
	Используй цикл for и функцию range().
	Входные данные
	Ничего не вводится.
	Выходные данные
	9 строк.
	Пример вывода (первые три строки)
	3 * 1 = 3
	3 * 2 = 6
	3 * 3 = 9
	...

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Решение
```
for i in range(1, 10):
    print(f'3 * {i} = {3 * i}')
```
Задача 3. Проверка номера телефона (только цифры)
Напиши программу, которая запрашивает строку и проверяет, является ли она корректным номером телефона.
Номер телефона считается корректным, если:
он состоит ровно из 11 символов;
все символы – цифры (0–9).
Если номер введён правильно – программа ничего не выводит.
Если номер введён неправильно – программа выводит строку «Неправильно набран номер».
Входные данные
Одна строка.
Выходные данные
Либо Неправильно набран номер, либо ничего.
Примеры
Ввод:
89271234567
Вывод: (пусто)
Ввод:
8927abc4567
Вывод:
Неправильно набран номер
Ввод:
892712345678
Вывод:
Неправильно набран номер
Примечание
Подсказки внутри input() не используй.
Решение
```
phone = input()
digits = '0123456789'
if len(phone) != 11:
    print('Неправильно набран номер')
else:
    for ch in phone:
        if ch not in digits:
            print('Неправильно набран номер')
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

	break
	Задача 4. Удаление гласных букв из строки
	Напиши программу, которая запрашивает строку на русском языке и удаляет из неё все гласные буквы (а, е, ё, и, о, у, ы, э, ю, я – в любом регистре).
	Программа должна вывести исходную строку и новую строку без гласных.
	Входные данные
	Одна строка.
	Выходные данные
	Две строки: сначала исходная, затем изменённая.
	Пример
	Ввод:
	Апельсин
	Вывод:
	Апельсин
	пльсн
	Примечание
	Подсказки внутри input() не используй.
	Решение
	text = input()
	vowels = 'аеёиоуыэюяАЕЁИОУЫЭЮЯ'
	result = ''
	for ch in text:
	if ch not in vowels:
	result += ch
	print(text)
	print(result)
	Задача 5. Сборка IP-адреса из списка
	Дан список:
	ip_parts = ['192', '168', '25', '41']
	Напиши программу, которая объединяет элементы этого списка в одну строку, разделяя их точкой (.).
	Выведи получившуюся строку.
	Входные данные
	Ничего не вводится.
	Выходные данные
	Одна строка – IP-адрес.
	Пример вывода
	192.168.25.41
	Решение

	ip_parts = ['192', '168', '25', '41'] print('.'.join(ip_parts))
Модуль 2, Обработка строк, Командный проект	
Количество академических часов	
Формы контроля	
Диагностические инструменты	
Показатели и критерии оценивания	
Модуль 3, Коллекции, Списки и кортежи	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи
Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполняет не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Марцинив Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Варианты заданий	Задача 1. Наименьшее из трёх чисел (условный оператор)
	Напиши программу, которая запрашивает три целых числа (каждое на отдельной строке) и выводит наименьшее из них.
	Входные данные
	Три целых числа, каждое на отдельной строке.
	Выходные данные
	Одно целое число – минимум.
	Пример
	Ввод:
	8
	3
	5
	Вывод:
	3
	Примечание
	Подсказки внутри input() не используй.
	Решение
	a = int(input())
	b = int(input())
	c = int(input())
	if a <= b and a <= c:
	  print(a)
	elif b <= a and b <= c:
	  print(b)
	else:
	  print(c)
	Задача 2. Целое или вещественное число?
	Напиши программу, которая запрашивает одно число (можно ввести как целое, так и дробное) и определяет, является ли оно целым или вещественным. Выведи соответствующую строку: «Целое число» или «Вещественное число».
	Входные данные
	Одно число.
	Выходные данные
	Строка Целое число или Вещественное число.
	Пример
	Ввод:
	7.25
	Вывод:
	Вещественное число
	Примечание

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Подсказки внутри input() не используй.
Решение

```
x = float(input())
if x == int(x):
    print('Целое число')
else:
    print('Вещественное число')
```

Задача 3. Калькулятор скидок
Программа предлагает пользователю выбрать действие:
1 – найти процент от числа
2 – вычесть процент из числа
Затем запрашивает число и процент (оба целые), выполняет выбранное действие и выводит результат (вещественное число).
Входные данные
Первая строка – номер действия (1 или 2).
Вторая строка – целое число.
Третья строка – целое число – процент.
Выходные данные
Сначала строка «1 - процент от числа, 2 - вычесть процент из числа», затем на новой строке – результат вычислений.
Пример
Ввод:

```
2
800
15
```

Вывод:

```
1 - процент от числа, 2 - вычесть процент из числа
680.0
```

Примечание
Подсказки внутри input() не используй.
Решение

```
print('1 - процент от числа, 2 - вычесть процент из числа')
choice = input()
num = int(input())
percent = int(input())
if choice == '1':
    print(num * percent / 100)
else:
    print(num - num * percent / 100)
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

	Задача 4. Игра «Угадай пароль» (цикл while)
	Программа ждёт, пока пользователь не введёт правильный пароль «Код». При каждой попытке выводится сообщение «Введите пароль:». После успешного ввода программа выводит сообщение «Доступ открыт» и завершается. Количество попыток не ограничено.
	Входные данные
	Строки (пароли) до тех пор, пока не будет введён «Код».
	Выходные данные
	Строки «Введите пароль:» перед каждым вводом, и «Доступ открыт» после правильного пароля.
	Пример работы
	Ввод:
	123
	qwerty
	Код
	Вывод:
	Введите пароль:
	Введите пароль:
	Введите пароль:
	Доступ открыт
	Примечание
	Подсказки внутри input() не используй.
	Решение
	while True:
	  print('Введите пароль:')
	  psw = input()
	  if psw == 'Код':
	  break
	print('Доступ открыт')
	Задача 5. Вывод чисел от 10 до 30, пропуская 15 и 25
	Напиши программу, которая выводит все целые числа от 10 до 30 включительно, но пропускает числа 15 и 25 (т.е. не выводит их). Каждое число выводится на отдельной строке.
	Входные данные
	Ничего не вводится.
	Выходные данные
	Числа от 10 до 30 (кроме 15 и 25), каждое на новой строке.
	Пример вывода (первые несколько строк)
	10
	11
	12
	13
	14

	16
	...
	30
	Примечание
	Подсказки внутри input() не используй.
	Решение
	for i in range(10, 31):
	  if i == 15 or i == 25:
	  continue
	  print(i)

Модуль 3, Коллекции, Ответственное использование ИИ-инструментов	
Количество академических часов	
Формы контроля	
Диагностические инструменты	
Показатели и критерии оценивания	
Модуль 3, Коллекции, Сортировка списков	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи
Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
академпер Маршева Виктория Вадимовна
серийный номер 01b93191c9412e804c6f27cc01aaaaa
срок действия 05.12.2025 - 05.03.2027

Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполняет не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)
Варианты заданий	Задача 1. Доступ к элементам списка Дан список: my_list = [12, 3, 45, 78, 9, 23] Напиши программу, которая выводит на экран второй элемент с начала и предпоследний элемент списка (каждый на отдельной строке). Входные данные Ничего не вводится. Выходные данные Два числа, каждое на новой строке. Пример вывода 3 9 Решение my_list = [12, 3, 45, 78, 9, 23] print(my_list[1]) # второй элемент (индекс 1) print(my_list[-2]) # предпоследний элемент Задача 2. Сумма чисел из строки Пользователь вводит строку целых чисел, разделённых пробелами. Напиши программу, которая преобразует строку в список чисел (тип int), вычисляет их сумму и выводит результат. Входные данные Строка целых чисел, разделённых пробелами. Выходные данные Одно целое число – сумма. Пример Ввод: 1 2 3 10 20 Вывод: 36 Примечание Подсказки внутри input() не используй. Решение s = input() numbers = list(map(int, s.split())) print(sum(numbers)) Задача 3. Подсчёт вхождений и удаление первого элемента по значению Дан список: my_list = [35, 156, 708, 35, 35, 61, 35] Напиши программу, которая: Подсчитывает, сколько раз в списке встречается число 35.

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

	Удаляет первое вхождение числа 35 из списка. Выводит сначала количество вхождений (целое число), затем изменённый список. Входные данные Ничего не вводится. Выходные данные Первая строка – количество вхождений числа 35. Вторая строка – изменённый список. Пример вывода 4 [156, 708, 35, 35, 61, 35] Решение my_list = [35, 156, 708, 35, 35, 61, 35] count_35 = my_list.count(35) my_list.remove(35) print(count_35) print(my_list) Задача 4. Сортировка списка товаров по длине названия Дан начальный список товаров: toys_list = ['конструктор лего', 'плюшевый медведь'] Программа запрашивает целое число n – количество новых товаров, затем n строк – названия товаров. Все новые товары добавляются в конец списка. После этого программа сортирует итоговый список по возрастанию длины названия (от самого короткого к самому длинному) и выводит отсортированный список. Входные данные Первая строка – целое число n. Далее n строк – названия товаров. Выходные данные Одна строка – отсортированный список. Пример Ввод: 3 дорога кукла игра Вывод: ['игра', 'кукла', 'дорога', 'конструктор лего', 'плюшевый медведь'] Примечание Подсказки внутри input() не используй. Решение

	toys_list = ['конструктор лего', 'плюшевый медведь']
	n = int(input())
	for _ in range(n):
	  toy = input()
	  toys_list.append(toy)
	print(sorted(toys_list, key=len))
	Задача 5. Фильтрация положительных чисел с помощью filter и lambda
	Дан список чисел:
	numbers = [1, -2, -3, 11, -5, 0, -7, -4, 99]
	Напиши программу, которая с помощью функции filter() и lambda-функции оставляет в списке только положительные числа (больше нуля). Выведи получившийся список.
	Входные данные
	Ничего не вводится.
	Выходные данные
	Список положительных чисел.
	Пример вывода
	[1, 11, 99]
	Решение
	numbers = [1, -2, -3, 11, -5, 0, -7, -4, 99]
	positive_numbers = list(filter(lambda x: x > 0, numbers))
	print(positive_numbers)

	Модуль 3, Коллекции, Словари и множества
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Маршеня Виктория Вадимовна
серийный номер 01b32191e94f2eb04e6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполняет не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)

Варианты заданий	1. Столицы стран (словарь)
	Создай словарь capitals, содержащий следующие пары «страна – столица»:
	Россия – Москва, Польша – Варшава, Грузия – Тбилиси, Турция – Анкара.
	Программа запрашивает у пользователя название страны (одна строка). Если страна есть в словаре – выведи её столицу. Если страны нет – выведи «Данные отсутствуют».
	Входные данные:
	Одна строка – название страны.
	Выходные данные:
	Одна строка – столица или сообщение об отсутствии.
	Пример ввода:
	Россия
	Пример вывода:
	Москва
	Решение:
	capitals = {
	'Россия': 'Москва',
	'Польша': 'Варшава',
	'Грузия': 'Тбилиси',
	'Турция': 'Анкара'
	}
	country = input()
	if country in capitals:
	print(capitals[country])
	else:
	print('Данные отсутствуют')
	2. Количество уникальных символов в строке
	Напиши программу, которая запрашивает строку и выводит количество различных символов (уникальных) в этой строке.
	Входные данные:
	Одна строка.
	Выходные данные:
	Одно целое число.
	Пример ввода:
	Hello, python!
	Пример вывода:
	12
	Решение:
	s = input()
	print(len(set(s)))

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

3. Обновление словаря (склад товаров)
Дан словарь с количеством товаров на складе:
stock = {'яблоки': 100, 'бананы': 333, 'груши': 200, 'апельсины': 300}
Программа запрашивает у пользователя строку (название товара) и целое число (новое количество).
Если товар уже есть в словаре – обнови его количество на введённое число.
Если товара нет – добавь новую пару «товар: количество».
Ввод повторяется до тех пор, пока не будет введено слово «стоп» (вместо названия товара).
После завершения ввода программа выводит итоговый словарь.
Входные данные:
Несколько строк: сначала строка с названием товара (или «стоп»), затем, если не «стоп», – строка с целым числом.
Выходные данные:
Словарь после всех изменений.
Пример работы (ввод):
бананы
150
лимоны
45
стоп
Пример вывода:
{'яблоки': 100, 'бананы': 150, 'груши': 200, 'апельсины': 300, 'лимоны': 45}
Решение:

```
stock = {'яблоки': 100, 'бананы': 333, 'груши': 200, 'апельсины': 300}
while True:
    item = input()
    if item == 'стоп':
        break
    quantity = int(input())
    stock[item] = quantity
print(stock)
```

4. Пересечение множеств (общие числа)
Напиши программу, которая запрашивает две строки целых чисел, разделённых пробелами, и выводит список чисел, которые встречаются в обеих строках (без повторений, в порядке возрастания).
Входные данные:
Две строки целых чисел, разделённых пробелами.
Выходные данные:
Список целых чисел в порядке возрастания.
Пример ввода:
3 2 1 5 4 3 8

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

	3 4 5 6 7 9
	Пример вывода:
	[3, 4, 5]
	Решение:
	set1 = set(map(int, input().split()))
	set2 = set(map(int, input().split()))
	common = set1 & set2
	print(sorted(common))
	5. Уникальность всех символов в строке
	Напиши программу, которая проверяет, все ли символы в строке уникальны (не повторяются).
	Если все символы встречаются только один раз – выведи «Все символы уникальны», иначе – «Есть повторяющиеся символы».
	Входные данные:
	Одна строка.
	Выходные данные:
	Сообщение – одна из двух указанных фраз.
	Пример ввода 1:
	abcdefg
	Пример вывода 1:
	Все символы уникальны
	Пример ввода 2:
	hello
	Пример вывода 2:
	Есть повторяющиеся символы
	Решение:
	s = input()
	if len(set(s)) == len(s):
	  print('Все символы уникальны')
	else:
	  print('Есть повторяющиеся символы')

Модуль 3, Коллекции, Проект	
Количество академических часов	
Формы контроля	
Диагностические инструменты	
Показатели и критерии оценивания	
Модуль 4, Модуль Turtle, Модули в Python	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи
Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполняет не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Варианты заданий	1. Обратный отсчёт с паузой
	Напиши программу, которая выводит обратный отсчёт от 10 до 0 включительно. Между выводом каждого числа должна быть пауза 0.3 секунды.
	Входные данные:
	Нет.
	Выходные данные:
	Числа от 10 до 0, каждое на новой строке.
	Пример вывода:
	10
	9
	8
	7
	6
	5
	4
	3
	2
	1
	0
	Решение:
	from time import sleep
	
	for i in range(10, -1, -1):
	  print(i)
	  sleep(0.3)
	2. Угадай число (от 1 до 30, 3 попытки)
	Напиши программу, которая загадывает случайное число от 1 до 30 (включительно) с помощью random.randint() и даёт игроку 3 попытки, чтобы его угадать. После каждого ввода программа выводит подсказку: «Загаданное число больше» или «Загаданное число меньше». Если игрок угадал, программа выводит «Ты угадал!». Если все попытки использованы, программа выводит «Попытки закончились! Загаданное число: X».
	Входные данные:
	От 1 до 3 целых чисел (каждое на новой строке).
	Выходные данные:
	Подсказки после каждого ввода, затем результат.
	Пример ввода:
	15
	20
	25
	Пример вывода (если загадано 18):
	Загаданное число больше
	Загаданное число больше

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Загаданное число меньше
Попытки закончились! Загаданное число: 18
Решение:

```
import random

secret = random.randint(1, 30)
for attempt in range(1, 4):
    guess = int(input())
    if guess == secret:
        print('Ты угадал!')
        break
    elif guess < secret:
        print('Загаданное число больше')
    else:
        print('Загаданное число меньше')
else:
    print(f'Попытки закончились! Загаданное число: {secret}')
```

3. Генерация лотерейного билета (6 чисел от 1 до 49)
Напиши программу, которая генерирует 6 случайных чисел от 1 до 49 (включительно) для лотерейного билета и выводит их на одной строке через запятую и пробел. Используй random.randint().
Входные данные:
Нет.
Выходные данные:
Строка с 6 числами через запятую и пробел.
Пример вывода:
7, 23, 15, 42, 8, 36
Решение:

```
import random

numbers = [random.randint(1, 49) for _ in range(6)]
print(', '.join(map(str, numbers)))
```

4. Случайный выбор победителя из списка
Напиши программу, которая запрашивает у пользователя имена участников через пробел, затем выводит сообщение «Определяем победителя...», ждёт 2 секунды и выводит случайное имя победителя с помощью random.choice().
Входные данные:
Одна строка с именами через пробел.
Выходные данные:
Строка «Определяем победителя...», затем имя победителя.
Пример ввода:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

	Анна Олег Мария Дмитрий Пример вывода: Определяем победителя... Мария Решение: import random import time names = input().split() print('Определяем победителя...') time.sleep(2) print(random.choice(names)) 5. Бросок двух кубиков (1–12) с суммой Напиши программу, которая моделирует бросок двух 12-гранных кубиков (каждый с числами от 1 до 12). Программа выводит значения первого и второго кубика, а затем их сумму. Входные данные: Нет. Выходные данные: Три строки: первый кубик, второй кубик, сумма. Пример вывода: Первый кубик: 7 Второй кубик: 4 Сумма: 11 Решение: import random dice1 = random.randint(1, 12) dice2 = random.randint(1, 12) print(f'Первый кубик: {dice1}') print(f'Второй кубик: {dice2}') print(f'Сумма: {dice1 + dice2}')
Модуль 4, Модуль Turtle, Модуль Turtle	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи
Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполняет не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Варианты заданий	1. Обратный отсчёт с измерением времени выполнения Напиши программу, которая выводит обратный отсчёт от 8 до 1 включительно, с паузой 0.4 секунды между числами. После завершения отсчёта программа должна вывести сообщение «Пуск!», а затем измерить и вывести время выполнения всей программы (в секундах, округлённое до целого). Используй time.sleep(), time.time(). Входные данные: Нет. Выходные данные: Числа от 8 до 1, затем «Пуск!» и время работы. Пример вывода: 8 7 6 5 4 3 2 1 Пуск! Время работы программы: 4 Решение: import time start = time.time() for i in range(8, 0, -1): print(i) time.sleep(0.4) print('Пуск!') end = time.time() print('Время работы программы:', round(end - start)) 2. Угадай число от 10 до 50 (4 попытки) Напиши программу, которая загадывает случайное целое число от 10 до 50 (включительно) и даёт игроку 4 попытки, чтобы угадать. После каждого ввода выводятся подсказки «Больше» или «Меньше». Если игрок угадал, программа выводит «Поздравляем!». Если попытки закончились, программа выводит «Конец игры! Загадано: X. Для ввода чисел используйinput(), для генерации –random.randint()`. Входные данные: От 1 до 4 целых чисел (каждое на новой строке). Выходные данные: Подсказки и результат. Пример ввода: 20 30 40

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

45
Пример вывода (если загадано 42):
Больше
Больше
Меньше
Меньше
Конец игры! Загадано: 42
Решение:

```
import random

secret = random.randint(10, 50)
for attempt in range(1, 5):
    guess = int(input())
    if guess == secret:
        print('Поздравляем!')
        break
    elif guess < secret:
        print('Больше')
    else:
        print('Меньше')
else:
    print(f'Конец игры! Загадано: {secret}')
```

3. Рисование правильного семиугольника с помощью turtle
Напиши программу, которая запрашивает у пользователя длину стороны (целое число) и рисует правильный семиугольник с заданной длиной стороны, используя модуль turtle. Цвет линии сделай синим ('blue'), толщину пера установи 5. Угол поворота вычисляется по формуле 360 / 7. Скорость черепахи установи 10.
Входные данные:
Одно целое число – длина стороны.
Выходные данные:
Рисуется семиугольник.
Пример ввода:
80
Решение:

```
from turtle import *

length = int(input())
color('blue')
pensize(5)
speed(10)
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

```
for _ in range(7):
    fd(length)
    left(360 / 7)
```

4. Рисование трёх окружностей с заливкой в разных координатах

Напиши программу, которая запрашивает три пары координат (x, y) – каждая пара вводится с новой строки (сначала x, потом y, затем следующая пара). В каждой точке нарисуй окружность радиусом 25 пикселей с заливкой (используй begin_fill() и end_fill()). Цвет окружностей: первая – красная, вторая – зелёная, третья – жёлтая. Используй penup() и pendown() для перемещения без следов.

Входные данные:

Шесть целых чисел: x1, y1, x2, y2, x3, y3 – каждое на отдельной строке.

Выходные данные:

Рисуются три закрашенные окружности.

Пример ввода:

```
-50
50
0
-50
50
20
```

Решение:

```
from turtle import *

def draw_circle(x, y, color_name):
    penup()
    goto(x, y)
    pendown()
    color(color_name)
    begin_fill()
    circle(25)
    end_fill()

x1 = int(input()); y1 = int(input())
x2 = int(input()); y2 = int(input())
x3 = int(input()); y3 = int(input())

draw_circle(x1, y1, 'red')
draw_circle(x2, y2, 'green')
draw_circle(x3, y3, 'yellow')
```

5. Класс Book (книга) с конструктором и методом info

Создай класс Book с конструктором __init__(), который принимает три параметра: title (название), author (автор), pages (количество страниц). В конструкторе сохрани их в атрибуты объекта. Добавь метод info(), который выводит строку в формате: «Название: {title}, Автор: {author}, Страниц: {pages}». Создай три объекта (например, для книг «Война и мир», «Преступление и наказание», «Мастер и Маргарита») и вызови для каждого метод info().

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

	Входные данные:
	Нет (данные заданы в коде).
	Выходные данные:
	Три строки с информацией о книгах.
	Пример вывода:
	Название: Война и мир, Автор: Толстой, Страниц: 1300
	Название: Преступление и наказание, Автор: Достоевский, Страниц: 550
	Название: Мастер и Маргарита, Автор: Булгаков, Страниц: 400
	Решение:
	class Book:
	  def __init__(self, title, author, pages):
	  self.title = title
	  self.author = author
	  self.pages = pages
	
	  def info(self):
	  print(f'Название: {self.title}, Автор: {self.author}, Страниц: {self.pages}')
	
	book1 = Book('Война и мир', 'Толстой', 1300)
	book2 = Book('Преступление и наказание', 'Достоевский', 550)
	book3 = Book('Мастер и Маргарита', 'Булгаков', 400)
	
	book1.info()
	book2.info()
	book3.info()

Модуль 4, Модуль Turtle, Кейс-задача	
Количество академических часов	
Формы контроля	
Диагностические инструменты	
Показатели и критерии оценивания	
Промежуточная аттестация (по каждому модулю)	
Первая программа, Модуль 1	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде. Практические задания на написание программного кода по предложенному алгоритму и описанию задачи. Задания на определение результата работы программного кода. Задания открытого типа.
Диагностические инструменты	Комплекты из 15 практических заданий по теме Не менее 50% заданий составляют практико-ориентированные задания на написание программного кода Не менее 20% заданий составляют задания открытого типа Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальное количество баллов — 15. Задание на написание программного кода считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью соответствовать ожидаемому результату для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Задания на исправление ошибок, определение результата работы программы и задания открытого типа считаются выполненными верно, если полученный ответ полностью соответствует эталонному ответу. Промежуточная аттестация считается пройденной при выполнении не менее 60% заданий (не менее 9 из 15 баллов). Прохождение промежуточной аттестации по модулю является обязательным условием для допуска обучающегося к итоговой аттестации.

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Варианты заданий	Задание 1 (написание кода) Напиши программу, которая выводит на экран три строки: – число 2027 и число 513 через пробел; – текст "result =" и число 15 через пробел; – текст "Я учу Python". Входные данные: Ничего не вводится. Выходные данные: Три строки в указанном формате. Пример вывода: 2027 513 result = 15 Я учу Python Возможное решение: print(2027, 513) print('result =', 15) print('Я учу Python') Задание 2 (исправление ошибок) Дана программа, которая должна создать переменную code со значением 257, вывести её, затем изменить значение на строку "Борис" и вывести Добро пожаловать, Борис: code = 257 print(code) code = Борис print('Добро пожаловать,', code) Найди и исправь ошибки, чтобы программа работала верно. Входные данные: Ничего не вводится. Выходные данные: Две строки: число и приветствие. Пример вывода: 257 Добро пожаловать, Борис Возможное решение: code = 257 print(code) code = 'Борис' print('Добро пожаловать,', code)

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Задание 3 (написание кода)
Напиши программу, которая запрашивает два вещественных числа и выводит их сумму.
Входные данные:
Два вещественных числа, каждое на отдельной строке.
Выходные данные:
Одно вещественное число – сумма.
5.67
2.33
Возможное решение:

```
a = float(input())
b = float(input())
print(a + b)
```

Задание 4 (исправление ошибок)
Дана программа, которая должна вычислять значение 5 * a**2 + 3 * b**(3+1) по введённым a и b:

```
a = int(input())
b = int(input())
print(5 * a**2 + 3 * b**(3+1))
```

Программа содержит ошибку в вычислениях (неправильный порядок операций). Исправь её так, чтобы она вычисляла именно 5 * a**2 + 3 * b**(3+1).
Входные данные:
Два целых числа, каждое на отдельной строке.
Выходные данные:
Одно целое число – результат выражения.
Пример ввода:
3
4
Пример вывода:
813
Возможное решение: (код уже верен, ошибки нет, но можно показать, что он правильный. В задании сказано "исправь", но если ошибки нет, то можно оставить как есть. Однако для единообразия можно внести опечатку, например, написать b**(3+1) как b**3+1. Тогда исправление будет очевидным. Сделаем так: в условии дадим неверный код:

```
a = int(input())
b = int(input())
print(5 * a**2 + 3 * b**3 + 1)   # неверно
```

Тогда правильное решение – убрать лишние сложения и поставить скобки:

```
a = int(input())
b = int(input())
print(5 * a**2 + 3 * b**(3+1))
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Задание 5 (написание кода)
В интернет-магазине остались товары категории «ноутбуки» по цене от 20 000 до 90 000 рублей включительно. Напиши программу, которая запрашивает категорию и максимальную цену пользователя, а затем выводит Товар есть - True или Товар есть - False.
Входные данные:
Первая строка – категория (строка), вторая – целое число (цена).
Выходные данные:
Строка вида Товар есть - True или Товар есть - False.
Пример ввода:
ноутбуки
50000
Пример вывода:
Товар есть - True
Возможное решение:

```
category = input()
price = int(input())
result = (category == 'ноутбуки') and (20000 <= price <= 90000)
print('Товар есть -', result)
```

Задание 6 (исправление ошибок)
Дана программа, которая определяет, чётное число или нечётное:
python

```
n = int(input())
if n % 2 = 0:
    print('Число', n, '- чётное')
else:
    print('Число', n, '- нечётное')
```

Найди и исправь ошибку.
Входные данные:
Одно целое число.
Выходные данные:
Строка с результатом.
Пример ввода:
text
7
Пример вывода:
text
Число 7 - нечётное
Возможное решение:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

```
python
n = int(input())
if n % 2 == 0:   # исправлено: == вместо =
    print('Число', n, '- чётное')
else:
    print('Число', n, '- нечётное')
```

Задание 7 (написание кода)
Билет в кино для взрослых (возраст ≥ 18) стоит 450 рублей, для детей (возраст < 18) – скидка 15%. Напиши программу, которая запрашивает возраст и выводит стоимость билета.
Входные данные:
Одно целое число – возраст.
Выходные данные:
Одно вещественное число – стоимость.
Пример ввода:
text
16
Пример вывода:
text
382.5

Возможное решение:
```
python
age = int(input())
price = 450.0
if age < 18:
    print(price * 0.85)
else:
    print(price)
```

Задание 8 (исправление ошибок)
Дана программа, которая ищет наибольшее из трёх чисел:
```
a = int(input())
b = int(input())
c = int(input())
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

```
if a >= b and a >= c:
    print(a)
elif b >= a and b >= c:
    print(b)
else:
    print(c)
```

Программа работает неверно, если все числа равны (выводит c, а не a). Исправь ошибку.
Входные данные:
Три целых числа, каждое на отдельной строке.
Выходные данные:
Одно целое число – наибольшее.
Пример ввода:
10
20
15
Пример вывода:
20
Возможное решение: (код уже корректен, но можно изменить условие, чтобы при равенстве выводилось любое. Однако в задании сказано "исправь", можно поменять на использование elif c >= в первом условии, а во втором использовать >= и т.д. Но проще оставить как есть, так как ошибки нет. Чтобы сделать задание осмысленным, дадим код с ошибкой, например, в первом условии > вместо >=, тогда при равенстве не сработает. Исправим:)
Дадим неверный код:
python

```
if a > b and a > c:
    print(a)
elif b > a and b > c:
    print(b)
else:
    print(c)
```

При равенстве выведет c. Исправленный код – c >=.

Задание 9 (написание кода)
Напиши программу, которая запрашивает целое число n и выводит в столбик все числа от 1 до n включительно.
Входные данные:
Одно целое число.
Выходные данные:
Числа от 1 до n, каждое на отдельной строке.

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Пример ввода:
6
Пример вывода:
1
2
3
4
5
6
Возможное решение:

```
n = int(input())
i = 1
while i <= n:
    print(i)
    i += 1
```

Задание 10 (ввод ответа)
Вопрос: Какой оператор в Python используется для полного выхода из цикла? Запиши ответ одним словом.
Ответ: break

Задание 11 (написание кода)
Напиши программу, которая запрашивает целое положительное число и вычисляет сумму его цифр.
Входные данные:
Одно целое число.
Выходные данные:
Одно целое число – сумма цифр.
Пример ввода:
345
Пример вывода:
12
Возможное решение:

```
n = int(input())
s = 0
while n > 0:
    s += n % 10
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

```
    n //= 10
print(s)
```

Задание 12 (ввод ответа)
Дана программа:

```
n = int(input())
s = 0
while n > 0:
    s += n % 10
    n //= 10
print(s)
```

Что выведет программа, если ввести число 345?
В ответ запиши одно целое число.
Ответ: 12

Задание 13 (написание кода)
Напиши программу, которая запрашивает целые числа до тех пор, пока не будет введён 0, а затем выводит сумму всех введённых чисел (без учёта завершающего нуля).
Входные данные:
Целые числа, каждое на отдельной строке. Последнее число – 0.
Выходные данные:
Одно целое число – сумма.
Пример ввода:
5
6
0
Пример вывода:
11
Возможное решение:

```
x = int(input())
total = 0
while x != 0:
    total += x
    x = int(input())
print(total)
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

	Задание 14 (ввод ответа)
	Вопрос: Какой тип данных в Python предназначен для хранения целых чисел? Запиши ответ одним словом.
	Ответ: int
	
	Задание 15 (написание кода)
	Компьютер загадал число 85. Напиши программу для игры «Угадай число»:
	Пользователь вводит числа.
	Если число меньше загаданного, выводится Загаданное число больше.
	Если больше – Загаданное число меньше.
	Если угадал – выводится Ты угадал с N попытки и программа завершается.
	Если пользователь вводит 888 – программа завершается без вывода.
	Входные данные:
	Целые числа, каждое на отдельной строке, пока не будет введено загаданное число или стоп-значение.
	Выходные данные:
	Подсказки или сообщение о победе.
	Пример ввода (победа):
	60
	100
	85
	Пример вывода:
	Загаданное число больше
	Загаданное число меньше
	Ты угадал с 3 попытки
	Возможное решение:
	secret = 85
	stop = 888
	count = 1
	n = int(input())
	while n != stop:
	  if n == secret:
	  print(f'Ты угадал с {count} попытки')
	  break
	  elif n < secret:
	  print('Загаданное число больше')
	  else:
	  print('Загаданное число меньше')
	  n = int(input())
	  count += 1

Шкала оценивания, нижнее значение	0
Шкала оценивания, верхнее значение	15
Шкала оценивания, минимальный проходной балл	9
Обработка строк, Модуль 2	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде. Практические задания на написание программного кода по предложенному алгоритму и описанию задачи. Задания на определение результата работы программного кода. Задания открытого типа.
Диагностические инструменты	Комплекты из 15 практических заданий по теме Не менее 50% заданий составляют практико-ориентированные задания на написание программного кода Не менее 20% заданий составляют задания открытого типа Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальное количество баллов — 15. Задание на написание программного кода считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью соответствовать ожидаемому результату для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Задания на исправление ошибок, определение результата работы программы и задания открытого типа считаются выполненными верно, если полученный ответ полностью соответствует эталонному ответу. Промежуточная аттестация считается пройденной при выполнении не менее 60% заданий (не менее 9 из 15 баллов). Прохождение промежуточной аттестации по модулю является обязательным условием для допуска обучающегося к итоговой аттестации.

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191e94f2cb04c6f27cc019aedad
срок действия 06.12.2025 - 06.12.2027

Варианты заданий	Задание 1 (написание кода) Напиши программу, которая запрашивает строку и два целых числа start и stop (каждое на отдельной строке), а затем выводит срез строки от индекса start (включительно) до индекса stop (не включительно). Входные данные: Строка, затем целое число start, затем целое число stop. Выходные данные: Одна строка – подстрока. Пример ввода: Привет, мир! 6 10 Пример вывода: т, м Возможное решение: s = input() start = int(input()) stop = int(input()) print(s[start:stop]) Задание 2 (исправление ошибок) Дана программа, которая должна вычислять сумму двух чисел с помощью функции calc: def calc(a, b) return a + b x = int(input()) y = int(input()) print(calc(x, y)) Программа содержит синтаксическую ошибку. Найди и исправь её так, чтобы она работала верно. Входные данные: Два целых числа, каждое на отдельной строке. Выходные данные: Одно целое число – сумма. Пример ввода: 5 8 Пример вывода: 13 Возможное решение: def calc(a, b): # добавлено двоеточие

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
    return a + b

x = int(input())
y = int(input())
print(calc(x, y))
```

Задание 3 (написание кода)

Напиши программу, которая запрашивает строку, удаляет из неё все пробелы и проверяет, является ли она палиндромом (без учёта регистра).

Если строка является палиндромом – вывести Палиндром, иначе – Это не палиндром.

Входные данные:

Одна строка.

Выходные данные:

Строка Палиндром или Это не палиндром.

Пример ввода:

А роза упала на лапу Азора

Пример вывода:

Палиндром

Возможное решение:

```
s = input().replace(' ', '')
if s.lower() == s[::-1].lower():
    print('Палиндром')
else:
    print('Это не палиндром')
```

Задание 4 (исправление ошибок)

Дана рекурсивная функция, которая должна вычислять факториал числа:

```
def factorial(n):
    if n == 0:
        return 1
    return factorial(n) * n

n = int(input())
print(factorial(n))
```

Однако программа работает неправильно (зацикливается). Найди и исправь ошибку.

Входные данные:

Одно целое число n ($n \geq 0$).

Выходные данные:

Одно целое число – факториал n.

Пример ввода:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

5
Пример вывода:
120
Возможное решение:

```
def factorial(n):
    if n == 0:
        return 1
    return factorial(n - 1) * n   # вызов с n-1
```

Задание 5 (написание кода)
Напиши программу, которая запрашивает строку на русском языке и удаляет из неё все гласные буквы (а, е, ё, и, о, у, ы, э, ю, я – в любом регистре). Остальные символы сохраняются.
Программа должна вывести сначала исходную строку, а затем – полученную строку без гласных (каждую на отдельной строке).
Входные данные:
Одна строка.
Выходные данные:
Две строки: исходная и изменённая.
Пример ввода:
Апельсин
Пример вывода:
Апельсин
пльсн
Возможное решение:

```
s = input()
vowels = 'аеёиоуыэюяАЕЁИОУЫЭЮЯ'
res = ''
for ch in s:
    if ch not in vowels:
        res += ch
print(s)
print(res)
```

Задание 6 (исправление ошибок)
Дана программа, которая должна выводить на экран квадраты чисел от 1 до 5:

```
for i in range(1, 5)
    print(i * i)
```

В программе допущена синтаксическая ошибка. Исправь её.
Входные данные:
Ничего не вводится.
Выходные данные:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Числа 1, 4, 9, 16 каждое на новой строке.
Пример вывода:
1
4
9
16
Возможное решение:
```
for i in range(1, 5):   # добавлено двоеточие
    print(i * i)
```
Задание 7 (написание кода)
Напиши программу, которая запрашивает целое число n и выводит таблицу умножения на n от 1 до 10 в формате:
n * i = результат.
Входные данные:
Одно целое число n.
Выходные данные:
10 строк в указанном формате.
Пример ввода:
3
Пример вывода:
3 * 1 = 3
3 * 2 = 6
...
3 * 10 = 30
Возможное решение:
```
n = int(input())
for i in range(1, 11):
    print(f'{n} * {i} = {n * i}')
```
Задание 8 (написание кода)
Напиши функцию arithmetic(a, b, op), которая принимает два числа и операцию ('+', '-', '*', '/'). Функция должна возвращать результат операции. Если операция неверна или деление на ноль – функция возвращает строку 'Ошибка'.
В основной программе запроси два числа и операцию, вызови функцию и выведи результат.
Входные данные:
Два числа (целых или вещественных), затем строка с операцией.
Выходные данные:
Результат операции (число) или строка 'Ошибка'.
Пример ввода:
65
76

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

*
Пример вывода:
4940
Возможное решение:

```
def arithmetic(a, b, op):
    if op == '+':
        return a + b
    elif op == '-':
        return a - b
    elif op == '*':
        return a * b
    elif op == '/':
        if b == 0:
            return 'Ошибка'
        return a / b
    else:
        return 'Ошибка'

a = float(input())
b = float(input())
op = input()
print(arithmetic(a, b, op))
```

Задание 9 (краткий ответ)
Какой метод строки в Python возвращает индекс первого вхождения подстроки, а если подстрока не найдена, возвращает -1?
В ответ запишите только название метода (без скобок).
Ответ: find
Задание 10 (краткий ответ)
Что называется рекурсией в программировании? Запишите определение одним предложением (не более 10 слов).
Ответ: функция, которая вызывает саму себя
Задание 11 (написание кода)
Напиши рекурсивную функцию sum_to(n), которая возвращает сумму всех целых чисел от 1 до n включительно. Не используй циклы.
В основной программе запроси число n и выведи результат.
Входные данные:
Одно целое число n ($n \geq 1$).
Выходные данные:
Одно целое число – сумма.
Пример ввода:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

5
Пример вывода:
15
Возможное решение:

```
def sum_to(n):
    if n == 1:
        return 1
    return n + sum_to(n - 1)

n = int(input())
print(sum_to(n))
```

Задание 12 (исправление ошибок)
Дана программа, которая должна заменять все вхождения подстроки "кот" на "пёс" в введённой строке:

```
s = input()
s = replace('кот', 'пёс')
print(s)
```

Найди и исправь ошибку.
Входные данные:
Одна строка.
Выходные данные:
Строка с заменой.
Пример ввода:
У меня есть кот и ещё один кот
Пример вывода:
У меня есть пёс и ещё один пёс
Возможное решение:

```
s = input()
s = s.replace('кот', 'пёс')   # метод вызывается от строки
print(s)
```

Задание 13 (написание кода)
Напиши программу, которая запрашивает строку и проверяет, является ли она номером телефона. Номер считается правильным, если он состоит ровно из 11 цифр (только цифры).
Если номер правильный – программа ничего не выводит. Если неправильный – выводит 'Неправильно набран номер'. Используй цикл for для проверки символов.
Входные данные:
Одна строка.
Выходные данные:
Либо ничего, либо строка 'Неправильно набран номер'.
Пример ввода:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

	89271234567
	Пример вывода:
	(пусто)
	Пример ввода 2:
	8927abc4567
	Пример вывода 2:
	Неправильно набран номер
	Возможное решение:
	phone = input()
	if len(phone) != 11:
	  print('Неправильно набран номер')
	else:
	  for ch in phone:
	  if ch not in '0123456789':
	  print('Неправильно набран номер')
	  break
	Задание 14 (краткий ответ)
	Какой метод строки разбивает строку на список подстрок по указанному разделителю?
	Запишите название метода без скобок.
	Ответ: split
	Задание 15 (написание кода)
	Напиши программу, которая запрашивает количество участников команды, затем их имена (каждое на отдельной строке).
	Программа должна сгенерировать логин из первых букв каждого имени (в нижнем регистре) и вывести его в формате: 'Ваш логин - ...'.
	Используй цикл for для обработки имён.
	Входные данные:
	Первая строка – целое число n (количество участников).
	Затем n строк – имена.
	Выходные данные:
	Одна строка с логином.
	Пример ввода:
	3
	Анна
	Борис
	Света
	Пример вывода:
	Ваш логин - абс
	Возможное решение:

	n = int(input())
	login = ''
	for i in range(n):
	  name = input()
	  login += name[0].lowe0r()
	print('Ваш логин -', login

Шкала оценивания, нижнее значение	0
Шкала оценивания, верхнее значение	15
Шкала оценивания, минимальный проходной балл	9
Коллекции, Модуль 3	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде. Практические задания на написание программного кода по предложенному алгоритму и описанию задачи. Задания на определение результата работы программного кода. Задания открытого типа.
Диагностические инструменты	Комплекты из 15 практических заданий по теме Не менее 50% заданий составляют практико-ориентированные задания на написание программного кода Не менее 20% заданий составляют задания открытого типа Встроенная автоматизированная система проверки кода

Контур Крипто

владелец Маршеня Виктория Вадимовна

Документ подписан квалифицированной электронной подписью 04.08.2026

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальное количество баллов — 15. Задание на написание программного кода считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью соответствовать ожидаемому результату для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Задания на исправление ошибок, определение результата работы программы и задания открытого типа считаются выполненными верно, если полученный ответ полностью соответствует эталонному ответу. Промежуточная аттестация считается пройденной при выполнении не менее 60% заданий (не менее 9 из 15 баллов). Прохождение промежуточной аттестации по модулю является обязательным условием для допуска обучающегося к итоговой аттестации.

Варианты заданий	Задание 1 (написание кода)
	Напиши программу, которая запрашивает у пользователя строку целых чисел, разделённых пробелами, преобразует её в список целых чисел, а затем выводит сумму всех чисел и количество элементов в списке (каждое на отдельной строке).
	Входные данные:
	Строка целых чисел, разделённых пробелами.
	Выходные данные:
	Два числа: сначала сумма, затем количество элементов.
	Пример ввода:
	1 2 3 10 20
	Пример вывода:
	36
	5
	Возможное решение:
	numbers = list(map(int, input().split()))
	print(sum(numbers))
	print(len(numbers))
	
	Задание 2 (исправление ошибок)
	Дана программа, которая должна добавлять новый элемент в начало списка:
	my_list = [10, 20, 30, 40]
	my_list.append(5, 0)
	print(my_list)
	Программа содержит ошибку. Найди и исправь её так, чтобы список стал [5, 10, 20, 30, 40].
	Входные данные:
	Ничего не вводится.
	Выходные данные:
	Список.
	Пример вывода:
	[5, 10, 20, 30, 40]
	Возможное решение:
	my_list = [10, 20, 30, 40]
	my_list.insert(0, 5) # вместо append
	print(my_list)
	
	Задание 3 (написание кода)
	Напиши программу, которая запрашивает строку, содержащую слова, разделённые пробелами, и выводит самое длинное слово в этой строке. Если таких слов несколько, выведи первое из них.

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Входные данные:
Строка слов.
Выходные данные:
Одно слово – самое длинное.
Пример ввода:
яблоко груша апельсин
Пример вывода:
апельсин
Возможное решение:

```
words = input().split()
longest = max(words, key=len)
print(longest)
```

Задание 4 (исправление ошибок)
Дана программа, которая должна удалять из списка все вхождения числа 5:

```
numbers = [5, 1, 5, 3, 5, 7]
for num in numbers:
    if num == 5:
        numbers.remove(num)
print(numbers)
```

Программа работает неправильно (не удаляет все пятёрки). Найди и исправь ошибку.
Входные данные:
Ничего не вводится.
Выходные данные:
Список без пятёрок.
Пример вывода:
[1, 3, 7]
Возможное решение:

```
numbers = [5, 1, 5, 3, 5, 7]
while 5 in numbers:
    numbers.remove(5)
print(numbers)
```

(или использовать список comprehension)

Задание 5 (написание кода)

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Напиши программу, которая запрашивает у пользователя строку и создаёт словарь, где ключами являются символы строки, а значениями – количество их вхождений (регистр не учитывается). Выведи полученный словарь.
Входные данные:
Строка.
Выходные данные:
Словарь с частотами символов (только для символов, которые встречаются хотя бы один раз).
Пример ввода:
abracadabra
Пример вывода:
{'a': 5, 'b': 2, 'r': 2, 'c': 1, 'd': 1}
Возможное решение:

```
s = input().lower()
freq = {}
for ch in s:
    freq[ch] = freq.get(ch, 0) + 1
print(freq)
```

Задание 6 (краткий ответ)
Вопрос:
Какой метод словаря возвращает список всех ключей? Запиши название метода без скобок.
Ответ: keys

Задание 7 (написание кода)
Напиши программу, которая запрашивает у пользователя две строки целых чисел, разделённых пробелами. Программа должна найти и вывести общие числа (пересечение) в виде списка, отсортированного по возрастанию.
Входные данные:
Две строки целых чисел.
Выходные данные:
Список общих чисел (уникальные, по возрастанию).
Пример ввода:
3 2 1 5 4 3 8
3 4 5 6 7 9
Пример вывода:
[3, 4, 5]
Возможное решение:

```
set1 = set(map(int, input().split()))
set2 = set(map(int, input().split()))
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

```
common = sorted(set1 & set2)
print(common)
```

Задание 8 (исправление ошибок)
Дана программа, которая должна объединять два словаря в один:

```
dict1 = {'a': 1, 'b': 2}
dict2 = {'c': 3, 'd': 4}
dict1 = dict1 + dict2
print(dict1)
```

Программа содержит ошибку. Исправь её так, чтобы результатом был объединённый словарь.
Входные данные:
Ничего не вводится.
Выходные данные:
Словарь.
Пример вывода:
{'a': 1, 'b': 2, 'c': 3, 'd': 4}
Возможное решение:

```
dict1 = {'a': 1, 'b': 2}
dict2 = {'c': 3, 'd': 4}
dict1.update(dict2)
print(dict1)
```

Задание 9 (написание кода)
Напиши программу, которая запрашивает строку чисел, разделённых пробелами, и с помощью filter и lambda оставляет только нечётные числа. Выведи полученный список.
Входные данные:
Строка целых чисел.
Выходные данные:
Список нечётных чисел.
Пример ввода:
1 2 3 4 5 6 7
Пример вывода:
[1, 3, 5, 7]
Возможное решение:

```
numbers = list(map(int, input().split()))
odd_numbers = list(filter(lambda x: x % 2 != 0, numbers))
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

```
print(odd_numbers)
```

Задание 10 (краткий ответ)
Вопрос:
Какой оператор в Python используется для объединения двух множеств? Запиши символ.
Ответ: |

Задание 11 (написание кода)
Напиши программу, которая запрашивает у пользователя имя и возраст (через пробел) для нескольких человек, пока не будет введено слово «стоп». Данные сохраняются в словарь (имя – ключ, возраст – значение). После завершения ввода программа выводит имя самого старшего человека.
Входные данные:
Строки вида имя возраст, затем «стоп».
Выходные данные:
Имя самого старшего.
Пример ввода:
Анна 25
Иван 30
Мария 22
стоп
Пример вывода:
Иван
Возможное решение:

```
people = {}
while True:
    s = input()
    if s == 'стоп':
        break
    name, age = s.split()
    age = int(age)
    people[name] = age
oldest = max(people, key=people.get)
print(oldest)
```

Задание 12 (исправление ошибок)
Дана программа, которая должна считать количество уникальных символов в строке:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

	s = input() unique = set(s) print(len(unique)) Программа работает верно, но в ней допущена логическая ошибка: она считает все уникальные символы, включая пробелы. По условию пробелы не должны учитываться. Исправь программу так, чтобы пробелы игнорировались. Входные данные: Строка. Выходные данные: Количество уникальных символов без пробелов. Пример ввода: hello world Пример вывода: 8 # h,e,l,o,w,r,d – 7? на самом деле букв: h,e,l,o,w,r,d – 7, но в "hello world" есть пробел, без него уникальные: h,e,l,o,w,r,d = 7, плюс буква 'o' уже есть, итого 7. Возможное решение: s = input().replace(' ', '') unique = set(s) print(len(unique)) Задание 13 (написание кода) Напиши программу, которая запрашивает список целых чисел (ввод через пробел) и с помощью map и lambda возводит каждый элемент в квадрат. Выведи полученный список. Входные данные: Строка чисел. Выходные данные: Список квадратов чисел. Пример ввода: 1 4 6 Пример вывода: [1, 16, 36] Возможное решение: numbers = list(map(int, input().split())) squared = list(map(lambda x: x ** 2, numbers)) print(squared) Задание 14 (краткий ответ) Вопрос: Какой метод списка удаляет последний элемент и возвращает его значение? Запиши название метода без скобок.
	Ответ: pop Задание 15 (написание кода) Напиши программу, которая запрашивает у пользователя строку, состоящую из слов, разделённых пробелами, и выводит список этих слов в обратном порядке (сначала последнее, затем предпоследнее и т.д.). Входные данные: Строка слов. Выходные данные: Список слов в обратном порядке. Пример ввода: один два три четыре Пример вывода: ['четыре', 'три', 'два', 'один'] Возможное решение: words = input().split() reversed_words = words[::-1] print(reversed_words)
Шкала оценивания, нижнее значение	0
Шкала оценивания, верхнее значение	15
Шкала оценивания, минимальный проходной балл	9
	Модуль Turtle, Модуль 4
Количество академических часов	1

Контур Крипто
владелец Маршеня Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 08.12.2025 - 08.03.2027

Формы контроля	Практические задания на исправление ошибок в готовом программном коде. Практические задания на написание программного кода по предложенному алгоритму и описанию задачи. Задания на определение результата работы программного кода. Задания открытого типа.
Диагностические инструменты	Комплекты из 15 практических заданий по теме Не менее 50% заданий составляют практико-ориентированные задания на написание программного кода Не менее 20% заданий составляют задания открытого типа Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальное количество баллов — 15. Задание на написание программного кода считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью соответствовать ожидаемому результату для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Задания на исправление ошибок, определение результата работы программы и задания открытого типа считаются выполненными верно, если полученный ответ полностью соответствует эталонному ответу. Промежуточная аттестация считается пройденной при выполнении не менее 60% заданий (не менее 9 из 15 баллов). Прохождение промежуточной аттестации по модулю является обязательным условием для допуска обучающегося к итоговой аттестации.
Варианты заданий	Задание 1 (написание кода) Напиши программу, которая: выводит на экран сообщение «Программа запущена»; ждёт 2 секунды (используй time.sleep); затем выводит сообщение «Текущее время:» и на следующей строке – текущее время в формате часы:минуты:секунды (например, 14:35:07). Используй time.localtime() и strftime() для форматирования. Входные данные: Ничего не вводится. Выходные данные: Три строки. Пример вывода: Программа запущена Текущее время: 14:35:07 Возможное решение: import time print('Программа запущена') time.sleep(2) print('Текущее время:') current = time.localtime() print(time.strftime('%H:%M:%S', current)) Задание 2 (написание кода) Напиши программу для игры «Угадай число». Компьютер загадывает случайное целое число от 1 до 50. У игрока есть 5 попыток. После каждого ввода программа выводит подсказку «Загаданное число больше» или «Загаданное число меньше». Если игрок угадал – выводится «Победа! Вы угадали с N попытки» и игра завершается. Если за 5 попыток не угадал – выводится «Проигрыш. Было загадано число X. Используй random.randint и random.seed(1)` для воспроизводимости. Входные данные: Целые числа (догадки) – не более 5 строк. Выходные данные: Подсказки и сообщение о результате. Пример ввода: 25 30 40 45 50 Пример вывода: Загаданное число больше

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
Загаданное число больше
Загаданное число больше
Загаданное число больше
Загаданное число больше
Проигрыш. Было загадано число 7
```

Возможное решение:

```
import random
random.seed(1)
secret = random.randint(1, 50)
for i in range(1, 6):
    guess = int(input())
    if guess == secret:
        print(f'Победа! Вы угадали с {i} попытки')
        break
    elif guess < secret:
        print('Загаданное число больше')
    else:
        print('Загаданное число меньше')
else:
    print(f'Проигрыш. Было загадано число {secret}')
```

Задание 3 (написание кода)

Напиши программу, которая с помощью модуля turtle рисует закрашенный квадрат со стороной 120 пикселей. Цвет заливки – жёлтый, цвет пера – красный, толщина пера – 3. После рисования черепаха должна быть скрыта. Используй команды: begin_fill(), end_fill(), color(), pensize(), forward(), left(), hideturtle().

Входные данные:

Ничего не вводится.

Выходные данные:

Графическое окно с квадратом. На платформе проверяется наличие всех команд и корректность синтаксиса.

Возможное решение:

```
from turtle import *
color('red')
pensize(3)
begin_fill()
color('yellow')
for _ in range(4):
    forward(120)
    left(90)
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

end_fill()
hideturtle()

Задание 4 (исправление ошибок)
Дана программа, которая должна выводить случайное целое число от 10 до 20:
import random
random.seed(1)
number = random.randint(10, 20)
print(number)
Программа работает верно. Однако в ней допущена синтаксическая ошибка: не хватает двоеточия после импорта (это не ошибка). На самом деле в условии можно дать неверный код, например, забыли import random или использовали randint без префикса.
Дадим неверный код:
from random import randint
random.seed(1)
number = randint(10, 20)
print(number)
Здесь ошибка: random.seed не определён, так как мы импортировали только randint, а не весь модуль random.
Исправь программу так, чтобы она работала.
Входные данные:
Ничего не вводится.
Выходные данные:
Одно целое число.
Пример вывода:
15 # зависит от seed, но seed(1) даёт конкретное число
Возможное решение:
import random
random.seed(1)
number = random.randint(10, 20)
print(number)
или
from random import randint, seed
seed(1)
number = randint(10, 20)
print(number)
Задание 5 (исправление ошибок)
Дана программа, которая должна нарисовать квадрат со стороной 100, но она работает неправильно. Найди и исправь ошибку.
from turtle import *

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

```
color('green')
pensize(3)
penup()
for _ in range(4):
    forward(100)
    left(90)
```

Входные данные:

Ничего не вводится.

Выходные данные:

На экране появляется квадрат.

Возможное решение:

```
from turtle import *
color('green')
pensize(3)
pendown()   # добавлено
for _ in range(4):
    forward(100)
    left(90)
```

Задание 6 (исправление ошибок)

Дана программа, которая должна создавать класс Book с конструктором и методом info(), но содержит ошибки:

```
class Book:
    def __init__(title, author, year):
        self.title = title
        self.author = author
        self.year = year
    def info():
        print(f'{self.title}, {self.author} ({self.year})')


book = Book('Война и мир', 'Толстой', 1869)
book.info()
```

Найди и исправь ошибки (их несколько).

Входные данные:

Ничего не вводится.

Выходные данные:

Строка с информацией о книге.

Пример вывода:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Война и мир, Толстой (1869)
Возможное решение:

```
class Book:
    def __init__(self, title, author, year):   # добавлен self
        self.title = title
        self.author = author
        self.year = year
    def info(self):                            # добавлен self
        print(f'{self.title}, {self.author} ({self.year})')


book = Book('Война и мир', 'Толстой', 1869)
book.info()
```

Задание 7 (написание кода)
Напиши программу, которая запрашивает у пользователя толщину пера (целое число) и рисует правильный треугольник со стороной 150 пикселей и заданной толщиной. Цвет пера – фиолетовый.
Входные данные:
Одно целое число – толщина пера.
Выходные данные:
Рисуется треугольник.
Пример ввода:
5
Возможное решение:

```
from turtle import *
w = int(input())
color('purple')
pensize(w)
for _ in range(3):
    forward(150)
    left(120)


```

Задание 8 (написание кода)
В программе задан список призов: prizes = ['наушники', 'книга', 'билет в кино', 'шоколадка', 'брелок']. Напиши программу, которая с помощью random.choice выбирает случайный приз, делает паузу в 1 секунду (используя time.sleep) и выводит сообщение: «Поздравляем! Вы выиграли: {приз}».
Входные данные:
Ничего не вводится.
Выходные данные:
Одна строка.

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Пример вывода:
Поздравляем! Вы выиграли: книга
Возможное решение:

```
import random
import time
prizes = ['наушники', 'книга', 'билет в кино', 'шоколадка', 'брелок']
time.sleep(1)
print(f'Поздравляем! Вы выиграли: {random.choice(prizes)}')
```

Задание 9 (написание кода)
Создай класс Student с конструктором, принимающим имя, возраст и средний балл. Добавь метод display(), который выводит строку в формате: «Имя: {имя}, Возраст: {возраст}, Балл: {балл}». В основной программе создай экземпляр класса с данными: «Анна», 16, 4.8 и вызови метод display().
Входные данные:
Ничего не вводится.
Выходные данные:
Одна строка.
Пример вывода:
Имя: Анна, Возраст: 16, Балл: 4.8
Возможное решение:

```
class Student:
    def __init__(self, name, age, score):
        self.name = name
        self.age = age
        self.score = score
    def display(self):
        print(f'Имя: {self.name}, Возраст: {self.age}, Балл: {self.score}')


student = Student('Анна', 16, 4.8)
student.display()
```

Задание 10 (исправление ошибок)
Программа должна рисовать красный круг радиусом 50, но она рисует чёрный круг. Найди и исправь ошибку.

```
import turtle
turtle.circle(50)
```

Входные данные:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Ничего не вводится.
Выходные данные:
Красный круг.
Возможное решение:

```
import turtle
turtle.color('red')
turtle.circle(50)
```

или

```
from turtle import *
color('red')
circle(50)
```

Задание 11 (исправление ошибок)
Дана программа, которая должна создать две черепашки и переместить их, но содержит ошибку. Найди и исправь синтаксическую ошибку в программе.

```
from turtle import *
t1 = Turtle()
t2 = Turtle()
t1.color('blue')
t2.color('green')
t1.goto(100 100)
t2.goto(-100, -100)
```

Возможное решение:

```
from turtle import *
t1 = Turtle()
t2 = Turtle()
t1.color('blue')
t2.color('green')
t1.goto(100, 100)   # добавлена запятая
t2.goto(-100, -100)
```

Задание 12 (исправление ошибок)
Дана программа, которая использует наследование от Turtle для создания класса Ball, но содержит ошибку:

```
from turtle import *
class Ball(Turtle):
    def __init__(self, x, y, color):
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

	__init__()
	self.penup()
	self.goto(x, y)
	self.color(color)
	self.shape('circle')
	
	ball = Ball(50, 50, 'red')
	В конструкторе пропущен вызов super().__init__() или Turtle.__init__(self). Исправь программу.
	Входные данные:
	Ничего не вводится.
	Выходные данные:
	Создаётся объект Ball (графически – круг).
	Возможное решение:
	from turtle import *
	class Ball(Turtle):
	def __init__(self, x, y, color):
	super().__init__() # или Turtle.__init__(self)
	self.penup()
	self.goto(x, y)
	self.color(color)
	self.shape('circle')
	
	ball = Ball(50, 50, 'red')
	
	Задание 13 (краткий ответ)
	Вопрос:
	Какой метод в модуле turtle используется для поворота черепахи на 90 градусов влево? Запиши название метода с правильным синтаксисом (например, forward(50)).
	Ответ: left(90)
	
	Задание 14 (краткий ответ)
	Вопрос:
	Как подключить функцию sleep из модуля time, чтобы использовать её как sleep(2) без префикса? Запиши строку импорта.
	Ответ: from time import sleep

	Задание 15 (краткий ответ) Вопрос: Какое ключевое слово используется в Python для создания класса? Запиши одним словом. Ответ: class
Шкала оценивания, нижнее значение	0
Шкала оценивания, верхнее значение	15
Шкала оценивания, минимальный проходной балл	9

Итоговый контроль/аттестация по ДОП	
Количество академических часов	2.00
Формы контроля	Защита итогового проекта, разработанного в рамках выполнения кейс-задачи, включает демонстрацию полученного результата и ответы на вопросы согласно утвержденному плану: Обоснование выбранной проблемы. Цель проекта и ожидаемый результат. Предложенное решение. Распределение задач между участниками. Демонстрация результата и его работоспособности. Анализ сильных сторон, недостатков и ограничений решения. Перспективы развития проекта.
Диагностические инструменты	Экспертный лист оценки проекта по установленным критериям Материалы итоговой защиты проекта

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Показатели и критерии оценивания	Проект оценивается по следующим критериям: соответствие заданию; проработка проекта; оригинальность решения; техническая сложность; практическая применимость; командная работа; защита (презентация) проекта. Каждый критерий оценивается по шкале от 0 до 3 баллов. Максимальное количество баллов — 21. Итоговая аттестация считается пройденной при наборе не менее 13 баллов (60% от максимального количества баллов). Подробное описание критериев, уровней оценивания и порядка выставления баллов представлено в Экспертном листе оценки проекта.
Варианты заданий	Защита итогового проекта, разработанного в ходе выполнения кейс-задачи, осуществляется в соответствии с требованиями, представленными в приложении к дополнительной образовательной программе (ДОП) в составе учебно-методического комплекса (УМК).
Шкала оценивания, нижнее значение	0.00
Шкала оценивания, верхнее значение	21.00
Шкала оценивания, минимальный проходной балл	13.00

Учебно-методические материалы

Наименование поля	Значение полей	Значение полей	Значение полей	Значение полей
Порядковый номер модуля	1	2	3	4
Методы, формы и технологии	Обучение реализуется на основе деятельностного, компетентностного и практико-ориентированного подходов. Освоение содержания модуля осуществляется через выполнение практических заданий, решение алгоритмических задач, разработку программ и реализацию проектов. Значительная часть учебного времени отводится практической деятельности, направленной на формирование навыков программирования и применение полученных знаний для решения практических задач. В образовательном процессе используются объяснение нового материала, демонстрация примеров, практические работы, решение задач, проектная деятельность, консультации, самостоятельная работа обучающихся и разбор типовых ошибок. Обучение осуществляется в групповой форме под руководством преподавателя. В зависимости от вида учебной деятельности используются фронтальная, индивидуальная и групповая формы организации работы обучающихся. В качестве образовательных технологий используются электронное обучение, дистанционные образовательные технологии, проектное обучение, технологии применения инструментов искусственного интеллекта, автоматизированная проверка заданий, средства визуализации алгоритмов и программного кода, а также современные цифровые инструменты разработки программ.	Обучение реализуется на основе деятельностного, компетентностного и практико-ориентированного подходов. Освоение содержания модуля осуществляется через выполнение практических заданий, решение алгоритмических задач, разработку программ и реализацию проектов. Значительная часть учебного времени отводится практической деятельности, направленной на формирование навыков программирования и применение полученных знаний для решения практических задач. В образовательном процессе используются объяснение нового материала, демонстрация примеров, практические работы, решение задач, проектная деятельность, консультации, самостоятельная работа обучающихся и разбор типовых ошибок. Обучение осуществляется в групповой форме под руководством преподавателя. В зависимости от вида учебной деятельности используются фронтальная, индивидуальная и групповая формы организации работы обучающихся. В качестве образовательных технологий используются электронное обучение, дистанционные образовательные технологии, проектное обучение, технологии применения инструментов искусственного интеллекта, автоматизированная проверка заданий, средства визуализации алгоритмов и программного кода, а также современные цифровые инструменты разработки программ.	Обучение реализуется на основе деятельностного, компетентностного и практико-ориентированного подходов. Освоение содержания модуля осуществляется через выполнение практических заданий, решение алгоритмических задач, разработку программ и реализацию проектов. Значительная часть учебного времени отводится практической деятельности, направленной на формирование навыков программирования и применение полученных знаний для решения практических задач. В образовательном процессе используются объяснение нового материала, демонстрация примеров, практические работы, решение задач, проектная деятельность, консультации, самостоятельная работа обучающихся и разбор типовых ошибок. Обучение осуществляется в групповой форме под руководством преподавателя. В зависимости от вида учебной деятельности используются фронтальная, индивидуальная и групповая формы организации работы обучающихся. В качестве образовательных технологий используются электронное обучение, дистанционные образовательные технологии, проектное обучение, технологии применения инструментов искусственного интеллекта, автоматизированная проверка заданий, средства визуализации алгоритмов и программного кода, а также современные цифровые инструменты разработки программ.	Обучение реализуется на основе деятельностного, компетентностного и практико-ориентированного подходов. Освоение содержания модуля осуществляется через выполнение практических заданий, решение алгоритмических задач, разработку программ и реализацию проектов. Значительная часть учебного времени отводится практической деятельности, направленной на формирование навыков программирования и применение полученных знаний для решения практических задач. В образовательном процессе используются объяснение нового материала, демонстрация примеров, практические работы, решение задач, проектная деятельность, консультации, самостоятельная работа обучающихся и разбор типовых ошибок. Обучение осуществляется в групповой форме под руководством преподавателя. В зависимости от вида учебной деятельности используются фронтальная, индивидуальная и групповая формы организации работы обучающихся. В качестве образовательных технологий используются электронное обучение, дистанционные образовательные технологии, проектное обучение, технологии применения инструментов искусственного интеллекта, автоматизированная проверка заданий, средства визуализации алгоритмов и программного кода, а также современные цифровые инструменты разработки программ.
Методические разработки (представлены в качестве приложения к ДОП)	Методические разработки модуля представлены в виде приложений к ДОП и включают описание модуля, поурочные планы учебных занятий, методические рекомендации по организации и проведению практических занятий, индивидуальных и командных проектов, самостоятельной работы обучающихся, текущего контроля и промежуточной аттестации. В состав методических разработок входят рекомендации по проведению занятий, краткое изложение теоретического материала, методические указания по выполнению заданий, формы контроля, показатели и критерии оценивания, возможные решения, а также инструкции по организации образовательного процесса, работе с образовательной платформой и системой автоматической проверки заданий.	Методические разработки модуля представлены в виде приложений к ДОП и включают описание модуля, поурочные планы учебных занятий, методические рекомендации по организации и проведению практических занятий, индивидуальных и командных проектов, самостоятельной работы обучающихся, текущего контроля и промежуточной аттестации. В состав методических разработок входят рекомендации по проведению занятий, краткое изложение теоретического материала, методические указания по выполнению заданий, формы контроля, показатели и критерии оценивания, возможные решения, а также инструкции по организации образовательного процесса, работе с образовательной платформой и системой автоматической проверки заданий.	Методические разработки модуля представлены в виде приложений к ДОП и включают описание модуля, поурочные планы учебных занятий, методические рекомендации по организации и проведению практических занятий, индивидуальных и командных проектов, самостоятельной работы обучающихся, текущего контроля и промежуточной аттестации. В состав методических разработок входят рекомендации по проведению занятий, краткое изложение теоретического материала, методические указания по выполнению заданий, формы контроля, показатели и критерии оценивания, возможные решения, а также инструкции по организации образовательного процесса, работе с образовательной платформой и системой автоматической проверки заданий.	Методические разработки модуля представлены в виде приложений к ДОП и включают описание модуля, поурочные планы учебных занятий, методические рекомендации по организации и проведению практических занятий, индивидуальных и командных проектов, самостоятельной работы обучающихся, текущего контроля, промежуточной и итоговой аттестации, а также методические рекомендации по организации и проведению кейс-задачи. В состав методических разработок входят рекомендации по проведению занятий, краткое изложение теоретического материала, методические указания по выполнению заданий, формы контроля, показатели и критерии оценивания, возможные решения, материалы итоговой аттестации, а также инструкции по организации образовательного процесса, работе с образовательной платформой и системой автоматической проверки заданий.
Материалы модуля (представлены в качестве приложения к ДОП)	Материалы модуля представлены в виде приложений к ДОП и включают презентационные материалы, материалы практических занятий, индивидуальных и командных проектов, самостоятельной работы обучающихся. В состав материалов входят описания заданий, необходимые учебные и справочные материалы, а также дополнительные материалы и файлы, используемые при освоении содержания модуля.	Материалы модуля представлены в виде приложений к ДОП и включают презентационные материалы, материалы практических занятий, индивидуальных и командных проектов, самостоятельной работы обучающихся. В состав материалов входят описания заданий, необходимые учебные и справочные материалы, а также дополнительные материалы и файлы, используемые при освоении содержания модуля.	Материалы модуля представлены в виде приложений к ДОП и включают презентационные материалы, материалы практических занятий, индивидуальных и командных проектов, самостоятельной работы обучающихся. В состав материалов входят описания заданий, необходимые учебные и справочные материалы, а также дополнительные материалы и файлы, используемые при освоении содержания модуля.	Материалы модуля представлены в виде приложений к ДОП и включают презентационные материалы, материалы практических занятий, индивидуальных и командных проектов, самостоятельной работы обучающихся, также материалы кейс-задачи. В состав материалов входят описания заданий, необходимые учебные и справочные материалы, а также дополнительные материалы и файлы, используемые при освоении содержания модуля.

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Марценя Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Учебная литература	Мэтиз, Э. Изучаем Python: программирование игр, визуализация данных, веб-приложения. 3-е изд. / Эрик Мэтиз; пер. с англ. Е. П. Матвеев; ил. Л. Ю. Егорова. — Москва : Питер, 2022. — 512 с. Васильев, А. Н. Программирование на Python в примерах и задачах / Алексей Николаевич Васильев. — Москва : Бомбора, 2023. — 616 с. Любанович, Б. Простой Python. Современный стиль программирования. 2-е изд. / Билл Любанович; пер. с англ. И. А. Пальти, Е. В. Зазноба; ил. Н. Гринчик. — Москва : Питер, 2022. — 592 с.	Мэтиз, Э. Изучаем Python: программирование игр, визуализация данных, веб-приложения. 3-е изд. / Эрик Мэтиз; пер. с англ. Е. П. Матвеев; ил. Л. Ю. Егорова. — Москва : Питер, 2022. — 512 с. Васильев, А. Н. Программирование на Python в примерах и задачах / Алексей Николаевич Васильев. — Москва : Бомбора, 2023. — 616 с. Любанович, Б. Простой Python. Современный стиль программирования. 2-е изд. / Билл Любанович; пер. с англ. И. А. Пальти, Е. В. Зазноба; ил. Н. Гринчик. — Москва : Питер, 2022. — 592 с.	Мэтиз, Э. Изучаем Python: программирование игр, визуализация данных, веб-приложения. 3-е изд. / Эрик Мэтиз; пер. с англ. Е. П. Матвеев; ил. Л. Ю. Егорова. — Москва : Питер, 2022. — 512 с. Васильев, А. Н. Программирование на Python в примерах и задачах / Алексей Николаевич Васильев. — Москва : Бомбора, 2023. — 616 с. Любанович, Б. Простой Python. Современный стиль программирования. 2-е изд. / Билл Любанович; пер. с англ. И. А. Пальти, Е. В. Зазноба; ил. Н. Гринчик. — Москва : Питер, 2022. — 592 с.	Мэтиз, Э. Изучаем Python: программирование игр, визуализация данных, веб-приложения. 3-е изд. / Эрик Мэтиз; пер. с англ. Е. П. Матвеев; ил. Л. Ю. Егорова. — Москва : Питер, 2022. — 512 с. Васильев, А. Н. Программирование на Python в примерах и задачах / Алексей Николаевич Васильев. — Москва : Бомбора, 2023. — 616 с. Любанович, Б. Простой Python. Современный стиль программирования. 2-е изд. / Билл Любанович; пер. с англ. И. А. Пальти, Е. В. Зазноба; ил. Н. Гринчик. — Москва : Питер, 2022. — 592 с. Python для детей. Самоучитель по программированию / Джейсон Бриггс ; пер. с англ. Станислава Ломакина ; [науч. ред. Д. Абрамова]. — М. : Манн, Иванов и Фербер, 2017. — 320 с.

Основные данные ДОП: ПРОДВИНУТЫЙ УРОВЕНЬ

№ п/п	Наименование поля	Значение поля
1.	Название ДОП	Python Про (продвинутый уровень, 8–9 классы)
2.	Уровень обучения ДОП	ПРОДВИНУТЫЙ
3.	Объем/общая трудоемкость ДОП (общий объем освоения ДОП в академических часах)	146.00
4.	Объем каждого модуля в ак.ч.	Модуль1 - 36 Модуль2 - 36 Модуль3 - 36 Модуль4 - 38
5.	Объем недельной нагрузки обучающегося	4.00
6.	Дата начала и дата окончания обучения по ДОП (срок освоения ДОП)	С 2026-09-28 по 2027-05-31

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

7.	Кейс-задача	Название кейс-задачи: Регистрация на школьное мероприятие через VK-бота Описание проблемной ситуации: При проведении школьных мероприятий, олимпиад, соревнований и мастер-классов организаторам необходимо зарегистрировать участников и подготовить информацию для проведения мероприятия. Если регистрация выполняется вручную, приходится обрабатывать большое количество однотипных заявок, проверять корректность введённых данных, удалять повторные регистрации и готовить итоговые списки участников. По мере увеличения числа участников такая работа становится более трудоёмкой, возрастает вероятность ошибок, а также появляется необходимость быстро выгружать данные, формировать бейджи и защищать регистрацию от спама и автоматической отправки заявок. Цель решения: Разработать командное решение для автоматизации регистрации участников на мероприятие с помощью VK-бота. Решение должно позволять пользователю подать заявку через сообщения боту, а организаторам — получать, хранить и обрабатывать список участников. Команда самостоятельно определяет состав продукта и дополнительные возможности: проверку данных, защиту от автоматических заявок, генерацию бейджей, экспорт списка участников и другие функции. Контекст: Для регистрации участников сегодня используются сайты, онлайн-формы, мобильные приложения и чат-боты. Одним из наиболее удобных способов является использование VK-бота сообщества, поскольку большинство участников уже пользуются социальной сетью и могут пройти регистрацию в привычном интерфейсе без установки дополнительных приложений. Бот позволяет организовать пошаговый сценарий регистрации с любой необходимой логикой, автоматически проверять введённые данные, предоставлять подтверждение регистрации, а также реализовывать дополнительные возможности для организаторов, например просмотр заявок, поиск участников, экспорт данных, генерацию бейджей и другие функции. Вопросы и задания: Какую проблему вы выбрали для решения и почему считаете её важной? Какую цель вы поставили перед своим проектом и какого результата планируете достичь? Какое решение проблемы предложила ваша команда? Расскажите о распределении задач между участниками. Покажите результат работы и объясните, как он помогает решить выбранную проблему. Какие сильные стороны имеет ваше решение и какие его недостатки или ограничения вы можете выделить? Какие улучшения или дополнительные возможности можно реализовать в будущем? Возможный результат выполнения задания: Подготовлена защита кейс-задачи, включающая обоснование выбранной проблемы, описание предложенного решения, демонстрацию созданного прототипа, MVP или программного продукта, анализ сильных сторон и ограничений решения, обсуждение возможных направлений его развития и описание вклада участников команды в общий результат. Критерии оценки: Проблема сформулирована и обоснована — 1 балл. Цель и ожидаемый результат соответствуют выбранной проблеме — 1 балл. Предложено решение, направленное на устранение или уменьшение
		проблемы — 1 балл. Создан прототип, MVP или работоспособный программный продукт — 1 балл. Представлена демонстрация решения и объяснён принцип его работы — 1 балл. Проведён анализ решения: определены его сильные стороны, ограничения и возможные направления развития — 1 балл. Объяснено распределение задач между участниками команды и вклад каждого участника в общий результат — 1 балл. Каждый вопрос оценивается в 1 балл. Балл выставляется, если команда дала содержательный и аргументированный ответ на соответствующий вопрос и продемонстрировала результат работы. Кейс-задача считается выполненной успешно, если команда набрала не менее 5 баллов из 7 возможных. Ограничения и допущения: Решение должно быть реализовано с использованием языка программирования Python. Допускается использование любых библиотек, фреймворков, API, баз данных и других программных инструментов, если они помогают решить поставленную задачу. Технические требования к оборудованию и программному обеспечению: Для решения кейс-задачи не требуется специализированное оборудование или программное обеспечение.
8.	Выдаваемый по окончании обучения документ	Все Получатели поддержки, успешно прошедшие промежуточную аттестацию по модулю ДОП, получают Сертификат Провайдера об освоении модуля. Все Получатели поддержки, успешно прошедшие итоговый контроль/аттестацию по ДОП допускаются к процедуре ИТиСИ. Все Получатели поддержки, успешно прошедшие ИТиСИ, получают Документ об обучении (форма «Университет 2035»).

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Описание ДОП

№ п/п	Наименование поля	Значение поля

1.	Актуальность ДОП	В условиях стремительного развития цифровой экономики и активного внедрения информационных технологий во все сферы деятельности возрастает потребность в формировании у обучающихся современных цифровых компетенций, навыков программирования и разработки программного обеспечения. Одним из наиболее востребованных инструментов для их формирования является язык программирования Python, который сочетает простоту освоения, широкие возможности применения и востребованность в различных областях информационных технологий. Отличительными особенностями содержания программы являются практико-ориентированная направленность обучения, углубленное изучение языка программирования Python, объектно-ориентированного программирования, алгоритмов и структур данных, современных технологий разработки программного обеспечения, развитие навыков проектной деятельности посредством выполнения индивидуальных и командных проектов, а также подготовка к участию в олимпиадах, конкурсах и соревнованиях в сфере информационных технологий с учетом возрастных особенностей обучающихся, их уровня подготовки и образовательных потребностей. Новизна программы заключается в сочетании изучения языка программирования Python с использованием современных технологий искусственного интеллекта и инструментов разработки программного обеспечения. Программа способствует совершенствованию алгоритмического мышления, развитию цифровых компетенций, навыков ответственного использования ИИ-технологий и формированию мотивации к дальнейшему обучению и профессиональному развитию в сфере информационных технологий.
2.	Цель и задачи ДОП	Цель программы: Формирование у обучающихся 8–9 классов углубленных знаний и практических навыков программирования на языке Python, совершенствование алгоритмического мышления, развитие цифровых компетенций, навыков ответственного использования ИИ-технологий и формирование мотивации к дальнейшему обучению и профессиональному развитию в сфере информационных технологий через решение практико-ориентированных задач повышенной сложности и выполнение проектов. Задачи программы: Обучающие: сформировать навыки создания, запуска, тестирования и отладки программ на языке Python; сформировать углубленные знания языка программирования Python, объектно-ориентированного программирования, алгоритмов и структур данных; обучить применению современных технологий разработки программного обеспечения, включая работу с файлами, API, форматами JSON и CSV, регулярными выражениями, тестированием и системами контроля версий; познакомить с возможностями использования технологий искусственного интеллекта в процессе разработки программ; сформировать основы ответственного и безопасного использования информационных технологий. Развивающие: развивать алгоритмическое, логическое и критическое мышление; развивать навыки поиска, анализа и исправления ошибок, а также отладки программ; развивать навыки самостоятельной работы с документацией и чтения программного кода; развивать навыки проектной, исследовательской и командной деятельности при выполнении индивидуальных и командных проектов; развивать способность применять современные технологии программирования для решения практических задач. Воспитательные: формировать ответственное отношение к работе с данными, включая вопросы конфиденциальности, безопасности и целостности информации; воспитывать этичное использование технологий искусственного интеллекта, уважение к авторским правам и интеллектуальной собственности; формировать навыки эффективной коммуникации и конструктивного взаимодействия при выполнении командных проектов; способствовать формированию устойчивого интереса к программированию, а также мотивации к дальнейшему обучению и профессиональному развитию в сфере информационных технологий.

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

3.	Планируемые результаты	По итогам освоения программы обучающийся будет: Иметь представление: о возможностях языка программирования Python для разработки программного обеспечения; о современных технологиях разработки программного обеспечения; о возможностях использования технологий искусственного интеллекта в процессе разработки программ; о принципах ответственного и безопасного использования информационных технологий; о возможностях дальнейшего обучения и профессионального развития в сфере информационных технологий. Знать: язык программирования Python, объектно-ориентированное программирование, алгоритмы и структуры данных; современные технологии разработки программного обеспечения, включая работу с файлами, API, форматами JSON и CSV, регулярными выражениями, тестированием и системами контроля версий; возможности использования технологий искусственного интеллекта в процессе разработки программ. Понимать: принципы создания, запуска, тестирования и отладки программ на языке Python; принципы разработки современных программных решений с использованием объектно-ориентированного программирования, алгоритмов и структур данных; возможности и ограничения использования технологий искусственного интеллекта в процессе разработки программ; принципы ответственного и безопасного использования информационных технологий. Уметь: создавать, запускать, тестировать и отлаживать программы на языке Python; применять объектно-ориентированное программирование, алгоритмы и структуры данных при разработке программ; использовать современные технологии разработки программного обеспечения, включая работу с файлами, API, форматами JSON и CSV, регулярными выражениями, тестированием и системами контроля версий; применять современные технологии программирования для решения практических задач; выполнять индивидуальные и командные проекты. Владеть: навыками создания, запуска, тестирования и отладки программ на языке Python; навыками применения современных технологий разработки программного обеспечения; навыками поиска, анализа и исправления ошибок в программном коде; навыками ответственного использования технологий искусственного интеллекта; навыками проектной, исследовательской и командной деятельности при выполнении индивидуальных и командных проектов.
4.	Дополнительная информация о ДОП	Дополнительная информация отсутствует
5.	Дополнительно (в соответствующих отдельных графах) указывается следующая информация по ДОП:	
5.1.	Стоимость обучения по ДОП	35558
5.2.	Рекомендации по обучению на ДОП	При обучении по программе рекомендуется руководствоваться дополнительной общеобразовательной программой, соблюдать последовательность освоения учебного материала, требования к организации обучения, выполнять предусмотренные программой формы текущего контроля, промежуточной и итоговой аттестации, а также использовать учебно-методические материалы.

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

3. Комплекс организационно-педагогических условий реализации ДОП:

Учебный план

№ п/п	Элементы ДОП	Элементы модуля (темы, промежуточная аттестация)	Общее кол-во часов элемента модуля, ак.час	Из них:			
				теоретическая подготовка, ак.час	практическая работа, ак.час	самостоятельная работа и самоконтроль, ак.час	контроль/ аттестация, ак.час
1.	Модуль 1 Основы Python	Основы Python	12	1	7	3	1
2.		Использование ИИ-инструментов	2	0	1	1	0
3.		Коллекции	11	1	6	3	1
4.		Функции	7	0	4	2	1
5.		Проект	3	0	2	1	0
6.		Промежуточная аттестация	1	0	0	0	1
7.	Модуль 2 Объектно-ориентированное программирование (ООП)	Объектно-ориентированное программирование (ООП)	15	1	8	5	1
8.		Парадигмы ООП	14	1	8	4	1
9.		Командный проект	6	0	4	1	1
10.		Промежуточная аттестация	1	0	0	0	1
11.	Модуль 3 Алгоритмы и структуры данных	Структуры данных	9	0	5	3	1
12.		Алгоритмы поиска и сортировки	10	1	5	3	1
13.		Ответственное использование ИИ-инструментов	2	0	1	1	0
14.		Сложность алгоритмов	9	1	5	2	1
15.		Командный проект	5	0	4	1	0
16.		Промежуточная аттестация	1	0	0	0	1
17.	Модуль 4 WEB и API	Работа с файлами	5	0	2	2	1
18.		Использование библиотек	11	1	6	3	1
19.		WEB и API	7	1	3	2	1
20.		Проект	5	0	4	1	0
21.		Кейс-задача	7	0	5	2	0
22.		Промежуточная аттестация	1	0	0	0	1
Итоговый контроль/аттестация по ДОП			2.00	0	0	0	2.00
Итого по ДОП:			146	8	80	40	18
ИТиСИ			Объем часов ИТиСИ не входит в общую трудоемкость ДОП, но участие в ИТиСИ обязательно для всех успешно прошедших Итоговый контроль/аттестацию по ДОП				

Контур.Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Машеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Рабочая программа с описанием каждого модуля ДОП

Элементы ДОП	Элементы модуля	Содержание (единицы содержания теоретической подготовки, практической работы, самостоятельной работы, контроля по теме и промежуточной аттестации (вопросы, задания, задачи и пр.))	Виды образовательных мероприятий /деятельности (теоретическая подготовка, практическая работа, самостоятельная работа, аттестация/контроль)	Объем в ак.ч.
Модуль 1 Основы Python Модуль направлен на формирование у обучающихся базовых навыков программирования на языке Python, развитие алгоритмического мышления и освоение основных конструкций языка. В рамках модуля обучающиеся изучают переменные и типы данных (int, float, str, bool), базовые операции, условные конструкции (if, elif, else), циклы (for, while), функции (def, параметры и return), а также работу с модулями (import, стандартные модули math, random) и получают первый опыт использования технологий искусственного интеллекта в программировании. Также выполняется проект «Игра в города». Задачи модуля: сформировать навыки создания, запуска, тестирования и отладки программ на языке Python; познакомить с базовыми типами данных (int, float, str, bool) и операциями над ними; сформировать навыки использования условных конструкций (if, elif, else); сформировать навыки использования циклов for и while; познакомить с созданием функций (def, параметры, return); познакомить с работой со стандартными модулями (import, math, random); сформировать навыки решения задач с использованием строк, списков, словарей и множеств; познакомить с возможностями использования технологий искусственного интеллекта при разработке программ; развить навыки проектной деятельности при выполнении индивидуального проекта. Планируемые результаты обучения По итогам освоения модуля обучающийся будет: знать: основные типы данных языка Python (int, float, str, bool); правила использования переменных и операций над данными; синтаксис условных конструкций (if, elif, else); синтаксис циклов for и while;	Основы Python	Интерпретатор и среда разработки. Ввод и вывод. Переменные и типы данных. Арифметические операции. Логический тип данных. Условный оператор. Циклы for и while. Операторы управления циклом.	Теоретическая подготовка	1
		Решение задач на ввод и вывод данных. Решение задач на использование переменных и типов данных. Решение задач на арифметические вычисления. Разработка простых программ с условием.	Практическая работа	7
		Решение задач на ввод и вывод данных. Решение задач на использование переменных и типов данных. Решение задач на арифметические вычисления. Разработка простых программ с условием. Выполнение тестовых заданий на ввод и вывод данных, логический тип данных, знание условного оператора.. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	3
		Решение задач на ввод и вывод данных (input(), print()), использование переменных и арифметических операций. Выполнение заданий с условным оператором if-elif-else и логическими операторами and, or. Решение задач с циклами for и while, включая вывод последовательностей и обработку диапазонов. Решение задач на накопление суммы и поразрядную обработку чисел (сумма цифр, разбор числа на цифры), поиск и исправление ошибок в коде.	Текущий контроль	1
	Использование ИИ-инструментов		Теоретическая подготовка	0
		ИИ-инструменты для программирования. Возможности и ограничения ИИ. Использование ИИ-инструментов в работе программиста. Разработка программ с использованием ИИ-инструмента. Поиск и исправление ошибок в коде, сгенерированном ИИ.	Практическая работа	1
		Разработка программ с использованием ИИ-инструмента. Поиск и исправление ошибок в коде, сгенерированном ИИ. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	1
			Текущий контроль	0

Контур Крипто
владелец Маршеня Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

основы работы с функциями (def, параметры, return); назначение стандартных модулей (math, random); назначение технологий искусственного интеллекта в программировании. уметь: создавать программы на языке Python; использовать типы данных и операции над ними; применять условные конструкции и циклы при решении задач; создавать и использовать функции; использовать модули math и random при решении задач; работать со строками и базовыми коллекциями данных; использовать технологии искусственного интеллекта для анализа и улучшения программного кода; выполнять тестирование и отладку программ владеть: навыками базового программирования на Python; навыками работы с типами данных и конструкциями языка; навыками использования функций и модулей; навыками решения алгоритмических задач; навыками работы с циклами и условиями; навыками проектной деятельности при выполнении проекта.	Коллекции	Коллекции в Python: строки, кортежи, списки, множества, словари. Методы списков. Методы для работы со словарями. Особенности множеств. Создание множеств. Функции и методы работы с множествами. Операции над множествами.	Теоретическая подготовка	1
		Решение задач на обработку строк. Выполнение операций со списками. Использование встроенных функций со списками при написании программ. Решение задач с использованием методов списков. Использование словарей при решении задач. Использование математических операций над множествами при решении задач. Решение стандартных и нестандартных задач по программированию.	Практическая работа	6
		Решение задач на обработку строк. Выполнение операций со списками. Использование встроенных функций со списками при написании программ. Решение задач с использованием методов списков. Использование словарей при решении задач. Использование математических операций над множествами при решении задач. Решение стандартных и нестандартных задач по программированию. Выполнение тестовых заданий на знание строк, методов списков, словарей. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	3
		Решение задач на обработку строк и списков: разбиение строк (split()), поиск и преобразование элементов. Решение задач с множествами: удаление повторов, нахождение пересечений и других операций над наборами данных. Решение задач со словарями: создание и обновление словаря, подсчёт частот элементов/символов. Решение прикладных задач на оптимизацию вычислений: использование формул арифметической прогрессии и алгоритма Евклида, поиск и исправление ошибок в коде.	Текущий контроль	1
	Функции		Теоретическая подготовка	0
		Определение и вызов функций в Python. Итераторы и итерируемые объекты. Генераторы. Рекурсия. Понятие локальных и глобальных переменных. Область видимости переменных. Декораторы. Модули и пакеты. Решение задач с использованием функций. Поиск и исправление ошибок в программном коде. Разработка программ с использованием функций. Решение задач с использованием модулей стандартной библиотеки и сторонних модулей.	Практическая работа	4
		Решение задач с использованием функций. Поиск и исправление ошибок в программном коде. Разработка программ с использованием функций. Решение задач с использованием модулей стандартной библиотеки и сторонних модулей. Выполнение тестовых заданий на знание функций. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	2

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Марщеня Виктория Вадимовна

серийный номер 01ba2101e94f2cb04e6f27ce019eedad

срок действия 06.12.2025 - 06.03.2027

		Решение задач с использованием lambda и функций высшего порядка filter() и map() для отбора и преобразования данных. Решение задач на сортировку и поиск экстремума с применением sorted() и max() (в том числе с параметром key). Решение задач на комбинирование функциональных инструментов (all(), filter(), map()) для проверки условий над набором данных (например, проверка палиндромов). Выполнение практических заданий на применение модуля math для вычислений с использованием констант и функций стандартной библиотеки.	Текущий контроль	1
	Проект		Теоретическая подготовка	0
		Решение задач на обработку строк и приведение ввода к единому формату (регистр, пробелы, дефисы). Реализация правил игры «Игра в города»: проверка наличия города в списке, проверка повторов, проверка корректной начальной буквы (с учётом исключений «ц», «ы», «ь»). Организация игрового цикла с завершением по команде «Сдаюсь» и при отсутствии подходящих городов у компьютера. Реализация хода компьютера: выбор первого подходящего города по алфавиту, вывод города с заглавной буквы и подсказки «Вам город на букву …». Тестирование и отладка проекта на типовых и граничных случаях (ошибки в заданном порядке, повторные города, разные регистры, отсутствие вариантов).	Практическая работа	2
		Доработка проекта «Игра в города». Тестирование программы на различных наборах входных данных. Поиск и исправление ошибок в программном коде.	Самостоятельная работа	1
			Текущий контроль	0
	Промежуточная аттестация	Практические задания на исправление ошибок в готовом программном коде. Практические задания на написание программного кода по предложенному алгоритму и описанию задачи. Задания на определение результата работы программного кода. Задания открытого типа. Комплекты из 15 практических заданий по теме Не менее 50% заданий составляют практико-ориентированные задания на написание программного кода Не менее 20% заданий составляют задания открытого типа Встроенная автоматизированная система проверки кода	Промежуточная аттестация	1
Модуль 2 Объектно-ориентированное программирование (ООП) Модуль направлен на формирование у обучающихся навыков объектно-ориентированного программирования на языке Python, развитие алгоритмического мышления и освоение принципов построения программных систем. В рамках модуля обучающиеся изучают классы и объекты, конструктор __init__, методы классов с использованием self, специальные методы арифметических операций (__add__, __sub__ и другие) и сравнения (__eq__, __lt__, __gt__ и другие), принципы наследования с использованием super(), полиморфизма и инкапсуляции, знакомство с графическим интерфейсом, обработку исключений (try, except, finally), а также выполняют командный проект «Игра Жизнь» и получают опыт командной разработки.	Объектно-ориентированное программирование (ООП)	Понятие объекта и класса в ООП. Методы классов и объектов. Специальные методы арифметических операций. Специальные методы сравнения. Применение ООП при разработке игровых проектов.	Теоретическая подготовка	1
		Решение задач с использованием ООП. Создание программ с использованием классов и объектов в Python. Разработка программ с использованием ООП. Создание игры «Крестики-нолики».	Практическая работа	8
		Решение задач с использованием ООП. Создание программ с использованием классов и объектов в Python. Разработка программ с использованием ООП. Выполнение тестовых заданий на знание классов и объектов. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	5

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Задачи модуля сформировать навыки объектно-ориентированного программирования в Python; познакомить с классами и объектами, конструктором __init__ и использованием self; сформировать навыки создания и использования методов классов; познакомить со специальными методами арифметических операций (__add__, __sub__ и другие) и сравнения (__eq__, __lt__, __gt__ и другие); сформировать навыки применения наследования с использованием super(); познакомить с принципами полиморфизма и инкапсуляции; познакомить с созданием интерфейсов с использованием библиотеки tkinter; познакомить с обработкой исключений (try, except, finally); развить навыки разработки программ с использованием объектно-ориентированного подхода; развить навыки командной проектной деятельности при выполнении проекта. Планируемые результаты обучения По итогам освоения модуля обучающийся будет: знать: принципы объектно-ориентированного программирования; назначение классов, объектов, конструктора __init__; использование self в методах классов; специальные методы (__add__, __sub__, __eq__, __lt__, __gt__ и другие); принципы наследования и использования super(); основы полиморфизма и инкапсуляции; основы работы с tkinter (Label, Button, Entry); основы обработки исключений (try, except, finally). уметь: создавать классы и объекты в Python; использовать конструктор __init__ и методы классов; применять специальные методы арифметических операций и сравнения; использовать наследование и super() при создании классов; применять принципы полиморфизма и инкапсуляции; создавать простые графические интерфейсы с использованием tkinter; обрабатывать ошибки с использованием try, except, finally; разрабатывать объектно-ориентированные программы; выполнять командный проект. владеть: навыками объектно-ориентированного программирования в Python; навыками работы с классами, объектами и методами; навыками использования специальных методов; навыками применения наследования, полиморфизма и инкапсуляции; навыками работы с графическим интерфейсом tkinter; навыками обработки исключений; навыками командной разработки программных систем.		Решение задач на создание классов и объектов: атрибуты, методы, конструктор __init__. Выполнение заданий на строковое представление объектов (__str__), использование @staticmethod и @classmethod в прикладных примерах. Решение задач на перегрузку операторов (__add__, __sub__, __mul__, __truediv__) и методов сравнения для пользовательских классов. Поиск и исправление ошибок в ООП-коде (корректное использование self, возвращаемые значения, проверка типов/условий).	Текущий контроль	1
	Парадигмы ООП	Концепции объектно-ориентированного программирования: наследование, инкапсуляция и полиморфизм. Знакомство с библиотекой Tkinter. Понятие исключения.	Теоретическая подготовка	1
		Создание программ с использованием наследования и полиморфизма. Создание простых приложений с графическим интерфейсом. Написание кода программы с использованием обработки исключений.	Практическая работа	8
		Создание программ с использованием наследования и полиморфизма. Создание простых приложений с графическим интерфейсом. Выполнение тестовых заданий на знание концепций ООП. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	4
		Решение задач на наследование: создание классов-наследников, переопределение методов, использование super(). Выполнение заданий на множественное наследование и корректный вызов методов базовых классов. Решение задач на полиморфизм: реализация единого интерфейса и работа с объектами разных классов через общие методы. Решение задач на инкапсуляцию: приватные атрибуты, свойства @property, сеттеры с проверкой корректности данных, поиск и исправление ошибок в коде.	Текущий контроль	1
	Командный проект		Теоретическая подготовка	0
		Понятие проекта, этапы разработки проекта и распределение задач в команде. Планирование и распределение задач при выполнении проекта. Разработка проекта «Игра Жизнь». Объединение результатов работы участников, тестирование и отладка программы. Подготовка защиты проекта. Изучение дополнительных материалов по теме. Доработка проекта и устранение ошибок. Тестирование программы на различных наборах входных данных. Подготовка презентации и защиты проекта.	Практическая работа	4
		Понятие проекта, этапы разработки проекта и распределение задач в команде. Планирование и распределение задач при выполнении проекта. Разработка проекта «Игра Жизнь». Объединение результатов работы участников, тестирование и отладка программы. Подготовка защиты проекта. Изучение дополнительных материалов по теме. Доработка проекта и устранение ошибок. Тестирование программы на различных наборах входных данных. Подготовка презентации и защиты проекта.	Самостоятельная работа	1
		Решение задач на создание классов и объектов: атрибуты, методы, конструктор __init__. Решение задач на наследование и переопределение методов с использованием super(), применение множественного наследования и полиморфизма. Выполнение заданий на инкапсуляцию: приватные атрибуты, свойства @property, сеттеры с проверкой корректности данных. Решение задач с обработкой исключений: try/except, создание и использование собственных исключений, поиск и исправление ошибок в ООП-коде.	Текущий контроль	1

Контур.Крипто
владелец Маршева Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3191c94f2eb04c6f27ee019aedad
срок действия 05.12.2025 - 05.03.2027

	Промежуточная аттестация	Практические задания на исправление ошибок в готовом программном коде. Практические задания на написание программного кода по предложенному алгоритму и описанию задачи. Задания на определение результата работы программного кода. Задания открытого типа. Комплекты из 15 практических заданий по теме Не менее 50% заданий составляют практико-ориентированные задания на написание программного кода Не менее 20% заданий составляют задания открытого типа Встроенная автоматизированная система проверки кода	Промежуточная аттестация	1
Модуль 3 Алгоритмы и структуры данных Модуль направлен на развитие у обучающихся алгоритмического мышления и навыков работы со структурами данных в языке Python, освоение базовых и прикладных алгоритмов обработки данных. В рамках модуля обучающиеся изучают массивы, связные списки, стек, дек и очередь, алгоритмы сортировки (выбором, пузырьком, вставками, слиянием), быструю сортировку, алгоритмы поиска (линейный и двоичный поиск), оценку сложности алгоритмов (О-нотация), а также решают задачи на структуры данных и алгоритмы, получают опыт применения ИИ-инструментов и выполняют командный проект «Очередь задач». Задачи модуля сформировать навыки работы со структурами данных в Python (списки, связные списки, стек, дек, очередь); сформировать навыки применения алгоритмов сортировки (выбором, пузырьком, вставками, слиянием) и быстрой сортировки; познакомить с алгоритмами поиска (линейный и двоичный поиск); сформировать представление о сложности алгоритмов (О-нотация); развить навыки решения задач на структуры данных и алгоритмы; познакомить с ответственным использованием технологий искусственного интеллекта при решении алгоритмических задач; развить навыки проектной деятельности при выполнении командного проекта. Планируемые результаты обучения По итогам освоения модуля обучающийся будет: знать: основные структуры данных Python (списки, связные списки, стек, дек, очередь); способы реализации стека и очереди; алгоритмы сортировки (выбором, пузырьком, вставками, слиянием, быстрая сортировка); алгоритмы поиска (линейный, двоичный); основы оценки сложности алгоритмов (О-нотация); принципы ответственного использования ИИ-инструментов. уметь: использовать структуры данных для хранения и обработки информации; реализовывать стек и очередь с использованием Python; применять алгоритмы сортировки и поиска;	Структуры данных		Теоретическая подготовка	0
		Знакомство с основными структурами данных. Массивы и списки в Python как динамические структуры. Стек, очередь, дек. Задачи на применение структур данных. Создание текстового редактора с Undo и Redo.	Практическая работа	5
		Задачи на применение структур данных. Создание текстового редактора с Undo и Redo. Выполнение тестовых заданий на знание структур данных. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	3
		Решение задач на реализацию и применение связного списка (поиск элементов, удаление узлов). Выполнение заданий на работу со стеком (LIFO), очередью (FIFO) и деком (добавление/удаление элементов с разных концов). Решение задач на обработку последовательностей с использованием подходящей структуры данных (выбор структуры под задачу). Реализация алгоритма вычисления выражений в обратной польской записи (ОПЗ) с использованием стека и поиск/исправление ошибок в коде.	Текущий контроль	1
	Алгоритмы поиска и сортировки	Знакомство с основными алгоритмами сортировки. Быстрые сортировки. Алгоритмы поиска.	Теоретическая подготовка	1
		Реализация алгоритмов сортировки в Python. Решение практических задач, связанных с быстрой сортировкой данных. Разработка программ поиска.	Практическая работа	5
		Реализация алгоритмов сортировки в Python. Решение практических задач, связанных с быстрой сортировкой данных. Разработка программ поиска. Выполнение тестовых заданий на знание алгоритмов поиска и сортировки. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	3
		Решение задач на реализацию алгоритмов сортировки: «тяжёлым камнем», выбором, вставками (в т.ч. с кастомным сравнением). Выполнение заданий на анализ работы сортировок (корректность, устойчивость, обработка граничных случаев). Решение задач на реализацию двоичного поиска левой и правой границы в отсортированном массиве. Поиск и исправление ошибок в программном коде, связанном с сортировкой и двоичным поиском	Текущий контроль	1
	Ответственное использование ИИ-инструментов		Теоретическая подготовка	0
		Безопасность при использовании ИИ. Авторство и этика использования ИИ. Анализ решений, полученных от ИИ. Решение задач с использованием ИИ. Проверка кода, сгенерированного ИИ.	Практическая работа	1
		Решение задач с использованием ИИ. Проверка кода, сгенерированного ИИ. Выполнение тестовых заданий на использование ИИ. Изучение дополнительных материалов по теме.	Самостоятельная работа	1

Контур Крипто
владелец Марценя Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba2191e94f2eb04c6f07cc019aedad
срок действия 05.12.2025 - 05.03.2027

анализировать эффективность алгоритмов на базовом уровне (О-нотация); решать задачи на структуры данных и алгоритмы; использовать ИИ-инструменты для анализа и оптимизации кода; выполнять командный проект. владеть: навыками работы со структурами данных; навыками реализации базовых алгоритмов сортировки и поиска; навыками оценки сложности алгоритмов; навыками решения алгоритмических задач повышенной сложности; навыками проектной и командной деятельности; навыками применения ИИ-инструментов в процессе разработки решений.			Текущий контроль	0
	Сложность алгоритмов	Понятие сложности алгоритмов и асимптотической оценки (Big O): O(1), O(log n), O(n), O(n log n), O(n^2); влияние размера входных данных на время работы. Типовые источники сложности. Основные структуры данных и их операции: список/массив, стек (LIFO), очередь (FIFO), дек. Принципы выбора структуры данных под задачу. Базовые приёмы оптимизации алгоритмов. Правила анализа и проверки решений: оценка сложности по коду, учёт граничных случаев, тестирование на больших входных данных и поиск «узких мест».	Теоретическая подготовка	1
		Решение задач на реализацию очереди (FIFO) на односвязном списке с операциями O(1) и обработкой команд. Решение задач на оценку временной сложности алгоритмов (O(1), O(n), O(n log n), O(n^2)) и сравнение вариантов решений. Решение алгоритмических задач с выбором подходящей структуры данных (список, стек, очередь, дек) для повышения эффективности. Реализация базовых структур данных и операций над ними (стек/очередь/дек) и применение в прикладных задачах. Решение задач на оптимизацию вычислений (замена вложенных циклов на более эффективные подходы, использование накопления/предобработки). Поиск и исправление ошибок в программном коде, анализ граничных случаев и проверка корректности решений.	Практическая работа	5
		Решение задач на реализацию очереди (FIFO) на односвязном списке с операциями O(1) и обработкой команд. Решение задач на оценку временной сложности алгоритмов (O(1), O(n), O(n log n), O(n^2)) и сравнение вариантов решений. Решение алгоритмических задач с выбором подходящей структуры данных (список, стек, очередь, дек) для повышения эффективности. Реализация базовых структур данных и операций над ними (стек/очередь/дек) и применение в прикладных задачах. Решение задач на оптимизацию вычислений (замена вложенных циклов на более эффективные подходы, использование накопления/предобработки). Поиск и исправление ошибок в программном коде, анализ граничных случаев и проверка корректности решений. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	2
		Решение задач на реализацию очереди на связном списке с операциями push, pop, size. Решение задач на реализацию циклического сдвига дека с использованием методов push_back, pop_back, push_front. Решение задач на проверку правильной скобочной последовательности с использованием стека. Решение задач на поиск пары чисел с заданной суммой в отсортированном массиве с использованием алгоритма двух указателей. Решение задач на проверку подпоследовательности с использованием алгоритма двух указателей.	Текущий контроль	1
	Командный проект		Теоретическая подготовка	0
		Понятие проекта, этапы разработки проекта и распределение задач в команде. Планирование и распределение задач при выполнении проекта. Разработка проекта «Очередь задач». Объединение результатов работы участников, тестирование и отладка программы. Подготовка защиты проекта. Изучение дополнительных материалов по теме. Доработка проекта и устранение ошибок. Тестирование программы на различных наборах входных данных. Подготовка презентации и защиты проекта.	Практическая работа	4

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

		Понятие проекта, этапы разработки проекта и распределение задач в команде. Планирование и распределение задач при выполнении проекта. Разработка проекта «Очередь задач». Объединение результатов работы участников, тестирование и отладка программы. Подготовка защиты проекта. Изучение дополнительных материалов по теме. Доработка проекта и устранение ошибок. Тестирование программы на различных наборах входных данных. Подготовка презентации и защиты проекта.	Самостоятельная работа	1
			Текущий контроль	0
	Промежуточная аттестация	Практические задания на исправление ошибок в готовом программном коде. Практические задания на написание программного кода по предложенному алгоритму и описанию задачи. Задания на определение результата работы программного кода. Задания открытого типа. Комплекты из 15 практических заданий по теме Не менее 50% заданий составляют практико-ориентированные задания на написание программного кода Не менее 20% заданий составляют задания открытого типа Встроенная автоматизированная система проверки кода	Промежуточная аттестация	1
Модуль 4 WEB и API Модуль направлен на формирование у обучающихся практических навыков работы с файлами, внешними данными и веб-сервисами в языке Python, освоение принципов взаимодействия с API, обработки и анализа данных, а также основ веб-разработки и автоматизации. В рамках модуля обучающиеся изучают работу с файлами (open, read, write), форматы данных JSON и CSV (json, csv), работу с командной строкой (sys, os), обработку даты и времени (datetime), регулярные выражения (re), тестирование программ (unittest), основы работы с API (библиотека requests), парсинг данных (HTML-разбор), основы работы с библиотекой vk_api, систему контроля версий Git, а также выполняют проект и решают кейс-задачу «Регистрация на школьное мероприятие через VK-бота». Задачи модуля сформировать навыки работы с файлами (open(), read(), write()); познакомить с форматами данных JSON и CSV (json, csv); сформировать навыки работы с командной строкой (sys, os); познакомить с работой с датой и временем (datetime); сформировать навыки использования регулярных выражений (re); познакомить с основами тестирования программ (unittest); сформировать навыки работы с веб-API (requests); познакомить с основами парсинга веб-страниц; сформировать навыки использования библиотеки vk_api; познакомить с системой контроля версий Git; развить навыки разработки программ, взаимодействующих с внешними сервисами; развить навыки проектной деятельности при выполнении командного проекта; развить навыки решения прикладной кейс-задачи. Планируемые результаты обучения По итогам освоения модуля обучающийся будет:	Работа с файлами		Теоретическая подготовка	0
		Понятие файла и файлового пути. Открытие/закрытие файлов в Python: функция open(), режимы работы ('r', 'w', 'a', 'rb', 'wb'), контекстный менеджер with. Чтение и запись текстовых файлов: read(), readline(), readlines(), перебор файла в цикле, write(), writelines(), обработка строк (удаление \n, разбиение split()). Обработка ошибок при работе с файлами: try/except, типовые исключения (FileNotFoundError, PermissionError), проверка корректности данных. Формат JSON: структура данных (объекты/массивы), соответствие типам Python; модуль json, функции json.load()/dump(), json.loads()/dumps(), параметры ensure_ascii, indent. Формат CSV: табличные данные, разделитель, заголовки; модуль csv, csv.reader, csv.writer, DictReader, DictWriter, параметр newline='', выбор разделителя delimiter.Решение задач на чтение и запись текстовых файлов с использованием open() и конструкции with. Решение задач на обработку данных из файла: построчное чтение, разбор строк (strip(), split()), подсчёты и формирование отчёта/результата. Выполнение практических заданий на сохранение и загрузку данных в формате JSON (json.load/dump), преобразование между словарями/списками и JSON. Выполнение практических заданий на чтение и запись CSV-файлов. Поиск и исправление ошибок при работе с файлами и форматами данных тестирование на разных входных наборах.	Практическая работа	2
		Решение задач на чтение и запись текстовых файлов с использованием open() и конструкции with. Решение задач на обработку данных из файла: построчное чтение, разбор строк (strip(), split()), подсчёты и формирование отчёта/результата. Выполнение практических заданий на сохранение и загрузку данных в формате JSON (json.load/dump), преобразование между словарями/списками и JSON. Выполнение практических заданий на чтение и запись CSV-файлов. Поиск и исправление ошибок при работе с файлами и форматами данных тестирование на разных входных наборах. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	2

Контур Крипто
владелец Марцева Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3191c94f2eb046f27ce019aedad
срок действия 05.12.2025 - 05.03.2027

По итогам освоения модуля обучающийся будет: знать: принципы работы с файлами в Python (open, read, write); форматы данных JSON и CSV (json, csv); основы работы с командной строкой (sys, os); основы работы с датой и временем (datetime); основы регулярных выражений (re); принципы тестирования программ (unittest); основы работы с API (requests); основы парсинга веб-страниц; назначение Git как системы контроля версий; основы работы с библиотекой vk_api. уметь: работать с файлами и файловой системой; обрабатывать данные в форматах JSON и CSV; использовать sys и os для работы с окружением; работать с датой и временем через datetime; применять регулярные выражения (re) для обработки текста; создавать тесты с использованием unittest; получать данные через API (requests); выполнять базовый парсинг веб-страниц; использовать vk_api для взаимодействия с API ВКонтакте; использовать Git на базовом уровне; выполнять командный проект; решать прикладную кейс-задачу. владеть: навыками работы с файлами и внешними данными; навыками обработки данных в форматах JSON и CSV; навыками использования системных библиотек Python; навыками работы с API и веб-данными; навыками тестирования программ; навыками базовой веб-автоматизации; навыками использования Git; навыками разработки прикладных программных решений; навыками командной проектной деятельности.		Решение задач на чтение и запись текстовых файлов. Решение задач на обработку данных из стандартного ввода и формирование корректного вывода по условию. Выполнение практических заданий на сохранение и загрузку структурированных данных в форматах JSON и CSV. Решение задач на использование аргументов командной строки (sys.argv) для задания входных/выходных файлов и режимов работы программы, поиск и исправление ошибок в коде.	Текущий контроль	1
	Использование библиотек	Командная строка: назначение, структура команды, чтение аргументов запуска, проверка корректности аргументов, вывод сообщений об ошибках. Дата и время: получение текущей даты и времени, форматирование и разбор дат, вычисление разницы между датами, добавление и вычитание интервалов. Регулярные выражения: назначение, шаблоны и основные метасимволы, поиск, извлечение и замена фрагментов текста. Тестирование программ: цели тестирования, тест-кейсы и граничные случаи, модульное тестирование, автоматическая проверка результатов.	Теоретическая подготовка	1
		Решение задач на запуск программ с параметрами командной строки и выбор режима работы программы. Выполнение заданий на работу с датой и временем: форматирование, разбор и вычисления с датами. Решение задач на поиск, проверку и замену данных в тексте с использованием регулярных выражений. Разработка и запуск набора тестов для функций, проверка ожидаемых результатов. Поиск и исправление ошибок, тестирование решений на типовых и граничных случаях.	Практическая работа	6
		Решение задач на запуск программ с параметрами командной строки и выбор режима работы программы. Выполнение заданий на работу с датой и временем: форматирование, разбор и вычисления с датами. Решение задач на поиск, проверку и замену данных в тексте с использованием регулярных выражений. Разработка и запуск набора тестов для функций, проверка ожидаемых результатов. Поиск и исправление ошибок, тестирование решений на типовых и граничных случаях. Изучение дополнительных материалов по теме. Подготовка к текущему контролю по теме.	Самостоятельная работа	3
		Решение задач на составление регулярных выражений для поиска, извлечения и замены фрагментов текста. Выполнение заданий на обработку текстовых данных с применением регулярных выражений и проверкой корректности результата. Написание проверок с использованием assert для контроля корректности работы функций. Разработка и запуск автоматизированных тестов с использованием doctest и unittest, анализ результатов и исправление ошибок.	Текущий контроль	1
	WEB и API	Основы Web-взаимодействия: клиент и сервер, запрос и ответ, адрес ресурса, параметры, коды ответа. Введение в API: назначение, принципы работы, методы запросов, заголовки, токены доступа, ограничения. Парсинг: получение данных из веб-страниц и ответов API, извлечение нужной информации, очистка и преобразование данных.	Теоретическая подготовка	1
		Решение задач на формирование запросов к API и получение данных по заданным параметрам. Выполнение заданий на обработку ответов. Решение задач на парсинг текстовых страниц: поиск и извлечение информации, очистка и структурирование данных. Решение задач на обработку ошибок и нестандартных ситуаций при запросах и парсинге, проверка корректности данных. Поиск и исправление ошибок в программном коде, тестирование решений на разных входных данных.	Практическая работа	3

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Марьина Виктория Вадимовна
серийный номер 01ba3191c94f2cb04cbf27cc019aedad
срок действия 05.12.2025 - 05.03.2027

		Решение задач на формирование запросов к API и получение данных по заданным параметрам. Выполнение заданий на обработку ответов. Решение задач на парсинг текстовых страниц: поиск и извлечение информации, очистка и структурирование данных. Решение задач на обработку ошибок и нестандартных ситуаций при запросах и парсинге, проверка корректности данных. Поиск и исправление ошибок в программном коде, тестирование решений на разных входных данных.	Самостоятельная работа	2
		Решение задач на выполнение HTTP-запросов к REST API с использованием requests и обработку кодов ответа. Выполнение заданий на разбор и обработку данных в формате JSON, извлечение нужных полей и преобразование результата. Решение задач на парсинг HTML-страниц с использованием Beautiful Soup: поиск элементов, извлечение текста и атрибутов. Разработка и отладка простых Telegram-ботов и VK-ботов: обработка команд, формирование ответов, работа с внешними данными.	Текущий контроль	1
	Проект		Теоретическая подготовка	0
		Создание бота для сообщества Работа с библиотекой vk-api Создание репозитория и ведение истории проекта: инициализация, добавление файлов, коммиты, просмотр истории изменений. Работа с GitHub: публикация репозитория, отправка и получение изменений, совместная работа через ветки и pull request. Разрешение типовых ситуаций: обновление проекта, устранение конфликтов при слиянии, оформление сообщений коммитов.	Практическая работа	4
		Доработка проекта и устранение ошибок. Тестирование программы на различных наборах входных данных.	Самостоятельная работа	1
			Текущий контроль	0
	Кейс-задача		Теоретическая подготовка	0
		Понятие кейс-задачи, анализ проблемной ситуации и формулирование требований к решению. Планирование командной разработки VK-бота: распределение ролей, проектирование сценария регистрации и структуры данных участников. Разработка прототипа или MVP «Регистрация на школьное мероприятие через VK-бота» с проверкой данных, защитой от повторных заявок и сохранением результатов. Объединение результатов работы участников, тестирование и отладка бота, проверка типовых и граничных случаев. Подготовка защиты кейс-задачи: демонстрация решения, объяснение принципов работы, анализ сильных сторон, ограничений и направлений развития, описание вклада участников.	Практическая работа	5
		Доработка функциональности бота и устранение выявленных ошибок. Тестирование решения на различных сценариях регистрации, включая некорректный ввод и повторные заявки. Подготовка материалов для защиты: презентация, описание архитектуры и сценариев, план улучшений.	Самостоятельная работа	2
			Текущий контроль	0
	Промежуточная аттестация	Практические задания на исправление ошибок в готовом программном коде. Практические задания на написание программного кода по предложенному алгоритму и описанию задачи. Задания на определение результата работы программного кода. Задания открытого типа. Комплекты из 15 практических заданий по теме Не менее 50% заданий составляют практико-ориентированные задания на написание программного кода Не менее 20% заданий составляют задания открытого типа Встроенная автоматизированная система проверки кода	Промежуточная аттестация	1

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Марцюня Виктория Вадимовна

серийный номер 01b3191c94f2e204c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Итоговый контроль/аттестация	Итоговый контроль/аттестация по ДОП	Защита итогового проекта, разработанного в рамках выполнения кейс-задачи, включает демонстрацию полученного результата и ответы на вопросы согласно утвержденному плану: Обоснование выбранной проблемы. Цель проекта и ожидаемый результат. Предложенное решение. Распределение задач между участниками. Демонстрация результата и его работоспособности. Анализ сильных сторон, недостатков и ограничений решения. Перспективы развития проекта. Экспертный лист оценки проекта по установленным критериям Материалы итоговой защиты проекта	Итоговый контроль/аттестация	2.00

		Объем в ак.ч.	Объем в %
ИТОГО ПО ПРОГРАММЕ:	Теоретическая подготовка	8,00	5,48
	Практическая работа	80,00	54,79
	Самостоятельная работа	40,00	27,40
	Текущий контроль	12,00	8,22
	Промежуточная аттестация	4,00	2,74
	Итоговый контроль/аттестация	2.00	1,37
	Всего, ак.ч.:	**146**	

ИТиСИ	Объем часов ИТиСИ не входит в общую трудоемкость ДОП, но участие в ИТиСИ обязательно для всех успешно прошедших Итоговый контроль/аттестацию по ДОП

Календарный учебный график

№ п/п	Название ДОП	№ потока	Дата начала обучения по ДОП	№ модуля ДОП	Начало обучения по модулю ДОП	Календарный период (количество дней) длительности модуля	Дата окончания обучения по ДОП
1	Python Про (продвинутый уровень, 8–9 классы)	1	2026-09-28	1	2026-09-28	64	2027-05-31
				2	2026-11-16	79	
				3	2027-01-11	79	
				4	2027-03-01	92	

Сведения об условиях реализации ДОП (в том числе материально-техническое, информационное, кадровое обеспечение)

Материально-технические и информационные условия реализации ДОП

Наименование поля	Значение полей	Значение полей	Значение полей	Значение полей
Порядковый номер модуля	1	2	3	4
Перечень требований к оснащению помещений (для офлайн-формата)	Не заполняется	Не заполняется	Не заполняется	Не заполняется

Контур.Крипто
владелец Маршеня Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Наименование требуемого оборудования (для онлайн-формата)	Компьютер или ноутбук с доступом к сети Интернет. Процессор: не менее 2 физических ядер с тактовой частотой от 2,0 ГГц. Оперативная память: не менее 4 ГБ (рекомендуется 8 ГБ и более). Свободное место на накопителе: не менее 4 ГБ. Разрешение экрана: не менее 1366×768 пикселей. Устройства ввода: клавиатура и мышь или иные устройства, обеспечивающие полноценную работу с компьютером. Наушники или колонки. Микрофон. Веб-камера (рекомендуется для участия в онлайн-занятиях).	Компьютер или ноутбук с доступом к сети Интернет. Процессор: не менее 2 физических ядер с тактовой частотой от 2,0 ГГц. Оперативная память: не менее 4 ГБ (рекомендуется 8 ГБ и более). Свободное место на накопителе: не менее 4 ГБ. Разрешение экрана: не менее 1366×768 пикселей. Устройства ввода: клавиатура и мышь или иные устройства, обеспечивающие полноценную работу с компьютером. Наушники или колонки. Микрофон. Веб-камера (рекомендуется для участия в онлайн-занятиях).	Компьютер или ноутбук с доступом к сети Интернет. Процессор: не менее 2 физических ядер с тактовой частотой от 2,0 ГГц. Оперативная память: не менее 4 ГБ (рекомендуется 8 ГБ и более). Свободное место на накопителе: не менее 4 ГБ. Разрешение экрана: не менее 1366×768 пикселей. Устройства ввода: клавиатура и мышь или иные устройства, обеспечивающие полноценную работу с компьютером. Наушники или колонки. Микрофон. Веб-камера (рекомендуется для участия в онлайн-занятиях).	Компьютер или ноутбук с доступом к сети Интернет. Процессор: не менее 2 физических ядер с тактовой частотой от 2,0 ГГц. Оперативная память: не менее 4 ГБ (рекомендуется 8 ГБ и более). Свободное место на накопителе: не менее 4 ГБ. Разрешение экрана: не менее 1366×768 пикселей. Устройства ввода: клавиатура и мышь или иные устройства, обеспечивающие полноценную работу с компьютером. Наушники или колонки. Микрофон. Веб-камера (рекомендуется для участия в онлайн-занятиях).
Наименование требуемого программного обеспечения (для онлайн-формата)	Операционная система: Windows 10 и выше, Linux или macOS. Современный веб-браузер с поддержкой актуальных веб-стандартов (рекомендуются последние версии Google Chrome или Яндекс Браузера). Среда разработки OpenIDE с установленным плагином Python Community Edition (при необходимости выполнения практических заданий на локальном компьютере). Интерпретатор Python версии 3.10 и выше (при необходимости выполнения практических заданий на локальном компьютере). Для прохождения обучения достаточно использования образовательной платформы и современного веб-браузера. Установка интерпретатора Python и среды разработки рекомендуется для выполнения практических заданий на локальном компьютере и не является обязательной.	Операционная система: Windows 10 и выше, Linux или macOS. Современный веб-браузер с поддержкой актуальных веб-стандартов (рекомендуются последние версии Google Chrome или Яндекс Браузера). Среда разработки OpenIDE с установленным плагином Python Community Edition (при необходимости выполнения практических заданий на локальном компьютере). Интерпретатор Python версии 3.10 и выше (при необходимости выполнения практических заданий на локальном компьютере). Для прохождения обучения достаточно использования образовательной платформы и современного веб-браузера. Установка интерпретатора Python и среды разработки рекомендуется для выполнения практических заданий на локальном компьютере и не является обязательной.	Операционная система: Windows 10 и выше, Linux или macOS. Современный веб-браузер с поддержкой актуальных веб-стандартов (рекомендуются последние версии Google Chrome или Яндекс Браузера). Среда разработки OpenIDE с установленным плагином Python Community Edition (при необходимости выполнения практических заданий на локальном компьютере). Интерпретатор Python версии 3.10 и выше (при необходимости выполнения практических заданий на локальном компьютере). Для прохождения обучения достаточно использования образовательной платформы и современного веб-браузера. Установка интерпретатора Python и среды разработки рекомендуется для выполнения практических заданий на локальном компьютере и не является обязательной.	Операционная система: Windows 10 и выше, Linux или macOS. Современный веб-браузер с поддержкой актуальных веб-стандартов (рекомендуются последние версии Google Chrome или Яндекс Браузера). Среда разработки OpenIDE с установленным плагином Python Community Edition. Интерпретатор Python версии 3.10 и выше. Git 2.4 и выше. Для выполнения практических заданий требуется установленный интерпретатор Python версии 3.10 и выше, а также рекомендованная среда разработки.

Информационные условия реализации ДОП (заполняется по каждому модулю):

Наименование поля	Значение полей	Значение полей	Значение полей	Значение полей
Порядковый номер модуля	1	2	3	4
Электронные информационные ресурсы	При реализации программы могут использоваться следующие электронные информационные и образовательные ресурсы: https://pythontutor.com/ — визуализатор выполнения кода на Python https://docs.python.org/3/ — официальная документация по Python https://docs.python.org/3/library/ — официальная дркументация по библиотекам Python https://proproprogs.ru/python — ресурс с информацией о программировании на Python	При реализации программы могут использоваться следующие электронные информационные и образовательные ресурсы: https://pythontutor.com/ — визуализатор выполнения кода на Python https://docs.python.org/3/ — официальная документация по Python https://docs.python.org/3/library/ — официальная дркументация по библиотекам Python https://proproprogs.ru/python — ресурс с информацией о программировании на Python	При реализации программы могут использоваться следующие электронные информационные и образовательные ресурсы: https://pythontutor.com/ — визуализатор выполнения кода на Python https://docs.python.org/3/ — официальная документация по Python https://docs.python.org/3/library/ — официальная дркументация по библиотекам Python https://proproprogs.ru/python — ресурс с информацией о программировании на Python	При реализации программы могут использоваться следующие электронные информационные и образовательные ресурсы: https://docs.python.org/3/ — официальная документация по Python https://docs.python.org/3/library/ — официальная документация по библиотекам Python https://proproprogs.ru/python — ресурс с информацией о программировании на Python https://github.com/python273/vk_api — библиотека vk-api https://dev.vk.com/ru/reference — официальная документация VK API

Контур КриптоПро
владелец Маршеня Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3191c94f2eb04c6f27ce019aefad
срок действия 05.12.2025 - 05.03.2027

Наименование поля	Значение полей	Значение полей	Значение полей	Значение полей
Порядковый номер модуля	1	2	3	4
Электронные образовательные ресурсы	При реализации программы могут использоваться следующие электронные информационные и образовательные ресурсы: https://pythontutor.ru/ — сборник задач по программированию https://acmp.ru/ — сборник задач по программированию https://informatics.msk.ru/ — сборник задач по программированию https://leetcode.com/ — задачи по программированию	При реализации программы могут использоваться следующие электронные информационные и образовательные ресурсы: https://pythontutor.ru/ — сборник задач по программированию https://acmp.ru/ — сборник задач по программированию https://informatics.msk.ru/ — сборник задач по программированию https://leetcode.com/ — задачи по программированию	При реализации программы могут использоваться следующие электронные информационные и образовательные ресурсы: https://pythontutor.ru/ — сборник задач по программированию https://acmp.ru/ — сборник задач по программированию https://informatics.msk.ru/ — сборник задач по программированию https://leetcode.com/ — задачи по программированию	При реализации программы могут использоваться следующие электронные информационные и образовательные ресурсы: https://pythontutor.ru/ — сборник задач по программированию https://acmp.ru/ — сборник задач по программированию https://informatics.msk.ru/ — сборник задач по программированию https://leetcode.com/ — задачи по программированию https://trudvsem.ru/opendata/api — API Работа России https://dadata.ru/ — API проверки, исправления и обогащения данных https://yandex.ru/dev/speller/doc/ru/concepts/api-overview — Яндекс Спеллер API

Кадровое обеспечение

Сведения о работниках, привлекаемых к реализации и разработке ДОП

ФИО	Участие в реализации ДОП	Наименование основного места работы	Ученая степень и (или) ученое звание (при наличии)	Образование (наименование учебного заведения, полученная специальность, год окончания)	Основание для привлечения к реализации ДОП (трудовой договор, договор ГПХ, иное)	Информация о дополнительном профессиональном образовании по профилю преподаваемой дисциплины (за последние 3 года)	Опыт работы в области ИТ, ИИ или робототехники и (или) профессиональные достижения в области ИТ, ИИ или робототехники (награды, научные статьи, иное)	Отметка о полученном согласии на обработку персональных данных
строка	(разработка/ реализация)	строка	строка	строка	строка	строка	строка	да/ нет
Подъельников Лев Андреевич	Разработка	Учи.Дома Ведущий методист		Чувашский государственный педагогический университет имени И.Я. Яковлева 2010 Физико-математический, Учитель математики и информатики	Трудовой договор			Да
Филиппов Егор Игоревич	Реализация	Учи.Дома Преподаватель		Федеральное государственное автономное образовательное учреждение высшего образования “Самарский национальный исследовательский университет имени академика С.П. Королева” 2022 Бизнес-информатика	Иное: Самозанятость	Федеральное государственное автономное образовательное учреждение высшего образования “Самарский национальный исследовательский университет имени академика С.П. Королева”, Системы искусственного интеллекта, Системы компьютерного зрения, Python для анализа данных, 2022г		Да
Лютов Егор Вячеславович	Реализация	Учи.Дома Преподаватель		Федеральное государственное бюджетное образовательное учреждение высшего образования "Уфимский государственный авиационный технический университет" 2026 Авионика, энергетика и инфокоммуникации	Иное: Самозанятость			Да
Гнездилова Ангелина Дмитриевна	Реализация	Учи.Дома Преподаватель		Федеральное государственное бюджетное образовательное учреждение высшего образования “Юго-Западный государственный университет” 2022 Математическое обеспечение и администрирование информационных систем	Иное: Самозанятость			Да

Контур Крипто
владелец Маршеня Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
сертификат номер 04bc8191c96f0ab94cc6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Толпеев Иван Васильевич	Реализация	Учи.Дома Преподаватель		Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования "Краснодарский государственный университет культуры и искусств" 2015 Менеджер социально-культурной деятельности	Иное: ИП	Общество с ограниченной ответственностью "Учи.ру", Особенности обучения программированию на курсе Python Start + AI в рамках проекта "Код Будущего", 2024	Разработчик, обл. ИИ, сайты, база данных, crm, 3 года.	Да
Асеев Ринат Раифович	Реализация	Учи.Дома Преподаватель		Уральский государственный педагогический университет 2013 Информационные технологии в образовании	Иное: Самозанятость		Инженер - программист, сети, программирование, 18 лет	Да
Небольсин Анатолий Александрович	Реализация	Учи.Дома Преподаватель		Федеральное государственное бюджетное образовательное учреждение высшего образования "Воронежский государственный университет инженерных технологий" 2024 Информационная безопасность автоматизированных систем	Иное: Самозанятость		нейронка от яндекс, тестировщик, около 3-4 месяцев	Да
Семенников Владислав Дмитриевич	Реализация	Учи.Дома Преподаватель		Федеральное государственное бюджетное образовательное учреждение высшего образования “Саратовский национальный исследовательский государственный университет имени Н.Г. Чернышевского” 2022 Информационные системы и программирование	Иное: Самозанятость			Да
Шлыков Вадим Николаевич	Реализация	Учи.Дома Преподаватель		Негосударственное аккредитованное частное образовательное учреждение высшего образования "Академия маркетинга и социально-информационных технологий - ИМСИТ" 2016 Программирование в компьютерных системах	Иное: Самозанятость			Да
Комарова Екатерина Юрьевна	Реализация	Учи.Дома преподаватель		Государственное образовательное учреждение высшего профессионального образования “Московский государственный индустриальный университет” 2009 Математическое обеспечение и администрирование информационных систем	Иное: Самозанятость			Да
Рожков Арсений Владимирович	Реализация	Учи.Дома Преподаватель		Московский государственный технический университет им. Н. Э. Баумана 2022 Прикладная информатика	Иное: Самозанятость		6 мес, младший разработчик пайтон, инженер систем, обл. программирование, до этого c++ - разработка, схожая позиция 1 год.	Да

Формы аттестации и оценочные материалы (комплект контрольно-измерительных материалов, позволяющих определить достижение Получателями поддержки планируемых результатов обучения)

Наименование поля	Значение полей
Текущий контроль	
Модуль 1, Основы Python, Основы Python	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи
Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполняет не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
Владелец Маршеня Виктория Вадимовна
серийный номер 01be2101c04f2ab04c6f37ec010eedad
срок действия 05.12.2025 - 05.03.2027

Варианты заданий	Задача 1. Арифметические операции над двумя числами
	Напиши программу, которая запрашивает два целых числа (каждое на отдельной строке). Программа должна вывести:
	сумму чисел;
	разность (первое минус второе);
	произведение;
	частное от деления первого числа на второе (вещественное число), если второе число не равно нулю.
	Если деление невозможно, вместо частного вывести строку «Деление на ноль невозможно».
	Входные данные
	Два целых числа, каждое на отдельной строке.
	Выходные данные
	Четыре строки: сумма, разность, произведение, частное (или сообщение об ошибке).
	Пример
	Ввод:
	10
	3
	Вывод:
	13
	7
	30
	3.3333333333333335
	Примечание
	Подсказки внутри input() не используй.
	Решение
	a = int(input())
	b = int(input())
	print(a + b)
	print(a - b)
	print(a * b)
	if b != 0:
	  print(a / b)
	else:
	print('Деление на ноль невозможно')
	Задача 2. Проверка, является ли четырёхзначное число «зеркальным» (палиндромом)
	Четырёхзначное число называется палиндромом, если оно одинаково читается слева направо и справа налево (например, 1221). Напиши программу, которая запрашивает одно целое четырёхзначное число и выводит «Да», если число является палиндромом, и «Нет» в противном случае.
	Входные данные
	Одно целое число (1000–9999).
	Выходные данные

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Строка «Да» или «Нет».
Пример
Ввод:
2442
Вывод:
Да
Примечание
Подсказки внутри input() не используй.
Решение

```
n = int(input())
d1 = n // 1000
d2 = n // 100 % 10
d3 = n // 10 % 10
d4 = n % 10
if d1 == d4 and d2 == d3:
    print('Да')
else:
    print('Нет')
```

Задача 3. Сумма всех нечётных делителей числа
Напиши программу, которая запрашивает целое положительное число n и вычисляет сумму всех его нечётных делителей (делители, которые не делятся на 2). Выведи полученную сумму.
Входные данные
Одно целое положительное число.
Выходные данные
Одно целое число – сумма нечётных делителей.
Пример
Ввод:
18
Вывод:
13
(Делители числа 18: 1, 2, 3, 6, 9, 18. Нечётные: 1, 3, 9. Сумма = 13.)
Примечание
Подсказки внутри input() не используй.
Решение

```
n = int(input())
total = 0
for i in range(1, n + 1):
    if n % i == 0 and i % 2 != 0:
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

```
        total += i
print(total)
```

Задача 4. Количество чётных чисел среди введённых (с досрочным выходом)

Напиши программу, которая запрашивает целые числа до тех пор, пока не будет введён 0. Программа должна подсчитать, сколько среди введённых чисел было чётных (не считая завершающего нуля). Если пользователь ввёл хотя бы одно нечётное число, программа должна вывести «Встретилось нечётное число» и завершиться (не дожидаясь ввода остальных чисел). Если все числа до нуля были чётными, программа должна вывести их количество.

Входные данные

Целые числа (каждое на новой строке). Последнее число – 0.

Выходные данные

Сообщение или количество чётных чисел.

Примеры

Ввод:

```
2
4
6
0
```

Вывод:

```
3
```

Ввод:

```
2
3
5
0
```

Вывод:

```
Встретилось нечётное число
```

Примечание

Подсказки внутри input() не используй.

Решение

```
count_even = 0
while True:
    x = int(input())
    if x == 0:
        break
    if x % 2 != 0:
        print('Встретилось нечётное число')
        break
    count_even += 1
else:
    print(count_even)
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

	Задача 5. Произведение цифр числа, кратных 3
	Напиши программу, которая запрашивает целое положительное число и вычисляет произведение всех его цифр, которые делятся на 3. Если таких цифр нет, программа должна вывести 0.
	Входные данные
	Одно целое положительное число.
	Выходные данные
	Одно целое число – произведение цифр, кратных 3 (или 0).
	Пример
	Ввод:
	1326
	Вывод:
	18
	(Цифры: 1, 3, 2, 6. Кратные 3: 3 и 6. Произведение = 18.)
	Примечание
	Подсказки внутри input() не используй.
	Решение
	n = int(input())
	product = 1
	found = False
	while n > 0:
	  digit = n % 10
	  if digit % 3 == 0:
	  product *= digit
	  found = True
	  n //= 10
	if found:
	  print(product)
	else:
	  print(0)

Модуль 1, Основы Python, Использование ИИ-инструментов	
Количество академических часов	
Формы контроля	
Диагностические инструменты	
Показатели и критерии оценивания	
Модуль 1, Основы Python, Коллекции	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи
Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполняет не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Варианты заданий	Задача 1. Создание аббревиатуры из слов
	Напиши программу, которая запрашивает у пользователя количество слов n, затем сами слова (каждое на отдельной строке). Программа должна сформировать аббревиатуру из первых букв каждого слова, переведённых в верхний регистр, и вывести полученную строку. Если пользователь ввёл число 0, программа должна вывести пустую строку (без ошибок).
	Входные данные
	Первая строка – целое число n ($0 \le n \le 100$).
	Далее n строк – слова (могут содержать только буквы).
	Выходные данные
	Одна строка – аббревиатура (заглавными буквами).
	Пример
	Ввод:
	4
	международный
	день
	защиты
	детей
	Вывод:
	МДЗД
	Примечание
	Подсказки внутри input() не используй.
	Решение
	n = int(input())
	abbr = ''
	for _ in range(n):
	word = input()
	if word:
	abbr += word[0].upper()
	print(abbr)
	Задача 2. Разделение чисел на чётные и нечётные (списки)
	Напиши программу, которая запрашивает 10 целых чисел (каждое на отдельной строке) и формирует два списка:
	список чётных чисел (в порядке ввода);
	список нечётных чисел (в порядке ввода).
	Выведи оба списка: сначала чётные, затем нечётные (каждый список на отдельной строке в формате Python).
	Входные данные
	10 целых чисел (по одному на строке).
	Выходные данные
	Две строки: список чётных и список нечётных чисел.
	Пример
	Ввод:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

1
2
3
4
5
6
7
8
9
10
Вывод:
[2, 4, 6, 8, 10]
[1, 3, 5, 7, 9]
Примечание
Подсказки внутри input() не используй.
Решение

```
evens = []
odds = []
for _ in range(10):
    x = int(input())
    if x % 2 == 0:
        evens.append(x)
    else:
        odds.append(x)
print(evens)
print(odds)
```

Задача 3. Пересечение множеств чисел из трёх строк
Даны три строки, каждая содержит целые числа, разделённые пробелами (числа в каждой строке уникальны). Напиши программу, которая находит числа, присутствующие во всех трёх строках, и выводит их в порядке возрастания через пробел. Если таких чисел нет, выведи пустую строку.
Входные данные
Три строки с целыми числами через пробел.
Выходные данные
Одна строка с общими числами, отсортированными по возрастанию, через пробел.
Пример
Ввод:
2 5 8 12
5 8 15 20
8 3 5 10

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Вывод:
5 8
Примечание
Подсказки внутри input() не используй.
Решение
```
set1 = set(map(int, input().split()))
set2 = set(map(int, input().split()))
set3 = set(map(int, input().split()))
common = sorted(set1 & set2 & set3)
print(' '.join(map(str, common)))
```
Задача 4. Самое часто встречающееся слово (словарь)
Напиши программу, которая считывает строку, содержащую слова, разделённые пробелами, и определяет слово, которое встречается наибольшее число раз. Если таких слов несколько, выведи первое из них по порядку в тексте. Гарантируется, что хотя бы одно слово есть.
Входные данные
Строка со словами через пробел.
Выходные данные
Одно слово – самое частое (первое при равенстве).
Пример
Ввод:
яблоко груша банан груша апельсин банан груша
Вывод:
груша
Примечание
Подсказки внутри input() не используй.
Решение
```
words = input().split()
freq = {}
for w in words:
    freq[w] = freq.get(w, 0) + 1
max_count = max(freq.values())
for w in words:
    if freq[w] == max_count:
        print(w)
        break
```
Задача 5. Количество нулей в двоичной записи числа
Напиши программу, которая запрашивает целое положительное число n ($n \le 10^6$) и выводит количество нулей в его двоичной записи. Если число равно 0, то в его двоичной записи одна цифра «0», поэтому программа должна вывести 1.
Входные данные
Одно целое число n.

Выходные данные
Одно целое число – количество нулей в двоичной записи.
Пример
Ввод:
10
Вывод:
2
(Двоичная запись 10 = 1010, нулей два.)
Примечание
Подсказки внутри input() не используй.
Решение
```
n = int(input())
if n == 0:
    print(1)
else:
    zeros = 0
    while n > 0:
        if n % 2 == 0:
            zeros += 1
        n //= 2
    print(zeros)
```

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

	Модуль 1. Основы Python. Функции
Количество академических часов	1

Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи
Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполняет не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)
Варианты заданий	Задача 1. Отбор чисел по остатку от деления Напиши функцию filter_remainder(*args, remainder), которая принимает произвольное количество целых чисел (позиционные аргументы) и один именованный аргумент remainder – целое число от 0 до 9. Функция должна вернуть список тех чисел, у которых остаток от деления на 10 равен remainder. Используй внутри filter() и lambda. Входные данные Первая строка – целые числа через пробел. Вторая строка – значение remainder (целое число). Выходные данные Список отобранных чисел в порядке их исходного появления. Пример Ввод: 12 35 48 77 90 23 65 5 Вывод: [35, 65] Примечание Подсказки внутри input() не используй. Решение def filter_remainder(*args, remainder): return list(filter(lambda x: x % 10 == remainder, args)) numbers = list(map(int, input().split())) rem = int(input()) print(filter_remainder(*numbers, remainder=rem)) Задача 2. Сортировка слов: сначала по длине, затем по алфавиту Напиши программу, которая запрашивает строку слов, разделённых пробелами. Отсортируй слова следующим образом: сначала по длине слова (от коротких к длинным); если длины совпадают, то по алфавиту (лексикографически). Выведи отсортированные слова на одной строке через пробел. Используй sorted() и lambda. Входные данные Строка слов через пробел. Выходные данные Строка отсортированных слов через пробел. Пример Ввод: кот собака мышь слон енот лев Вывод: кот лев слон енот мышь собака

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Примечание
Подсказки внутри input() не используй.
Решение
```
words = input().split()
sorted_words = sorted(words, key=lambda w: (len(w), w))
print(' '.join(sorted_words))
```
Задача 3. Поиск числа с максимальной суммой цифр
Напиши программу, которая запрашивает строку, содержащую слова и целые числа, разделённые пробелами. Найди среди всех целых чисел то, у которого сумма цифр максимальна. Если таких чисел несколько, выведи первое по порядку. Гарантируется, что хотя бы одно число есть.
Используй max() с параметром key и filter() для отбора только чисел.
Входные данные
Строка из слов и чисел через пробел.
Выходные данные
Одно целое число (исходное, не сумма цифр).
Пример
Ввод:
abc 123 def 45 ghi 6789
Вывод:
6789
(Суммы цифр: 1+2+3=6, 4+5=9, 6+7+8+9=30)
Примечание
Подсказки внутри input() не используй.
Решение
```
parts = input().split()
numbers = list(filter(lambda x: x.isdigit(), parts))
result = max(numbers, key=lambda n: sum(map(int, n)))
print(result)
```
Задача 4. Проверка: все ли слова в строке – палиндромы
Условие
Напиши программу, которая запрашивает строку из слов, разделённых пробелами. Словом считается последовательность букв (русских или английских). Цифры и другие символы игнорируются – они не считаются словами и не участвуют в проверке. Проверь, что каждое слово (только буквенное) является палиндромом (читается одинаково слева направо и справа налево). Выведи True, если все слова-палиндромы, и False в противном случае. Если в строке нет ни одного слова (только цифры или символы), выведи True.
Используй all(), map(), filter() и lambda.
Входные данные
Строка слов и символов через пробел.
Выходные данные
True или False.
Примеры
Ввод:
шалаш казак 123 дед око

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

	Вывод:
	True
	Ввод:
	шалаш казак привет
	Вывод:
	False
	Примечание
	Подсказки внутри input() не используй.
	Решение
	words = input().split()
	alpha_words = list(filter(lambda w: w.isalpha(), words))
	if not alpha_words:
	  print(True)
	else:
	  result = all(map(lambda w: w == w[::-1], alpha_words))
	  print(result)
	Задача 5. Геометрические вычисления с модулем math
	Напиши программу, которая запрашивает радиус круга (вещественное число) и с помощью константы math.pi вычисляет:
	длину окружности (C = 2πR);
	площадь круга (S = πR²).
	Выведи оба значения на отдельных строках без округления (как есть).
	Входные данные
	Одно вещественное число – радиус.
	Выходные данные
	Два вещественных числа, каждое на новой строке.
	Пример
	Ввод:
	3
	Вывод:
	18.84955592153876
	28.274333882308138
	Примечание
	Подсказки внутри input() не используй.
	Решение
	import math
	r = float(input())
	c = 2 * math.pi * r

	s = math.pi * r ** 2 print(c) print(s)
Модуль 1, Основы Python, Проект	
Количество академических часов	
Формы контроля	
Диагностические инструменты	
Показатели и критерии оценивания	
Модуль 2, Объектно-ориентированное программирование (ООП), Объектно-ориентированное программирование (ООП)	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи
Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполняет не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Варианты заданий	1. Класс «Книга» с информацией о прочтении
	Создай класс Book с атрибутами: title (название), author (автор), pages (количество страниц), is_read (прочитана ли книга – bool, по умолчанию False). Реализуй:
	конструктор __init__(title, author, pages);
	метод read(), который отмечает книгу как прочитанную (is_read = True);
	метод info(), который выводит строку в формате:
	«{автор} — {название} ({страницы} стр.) — {"Прочитана" if is_read else "Не прочитана"}».
	В основной программе создай два экземпляра класса Book с данными, введёнными с клавиатуры (для каждой книги: название, автор, количество страниц). Для второй книги вызови метод read(). Выведи информацию о каждой книге.
	Входные данные:
	Четыре строки: название первой книги, автор первой книги, количество страниц первой книги, название второй книги, автор второй книги, количество страниц второй книги.
	Выходные данные:
	Две строки – результат вызова метода info().
	Пример ввода:
	Война и мир
	Толстой
	1425
	1984
	Оруэлл
	328
	Пример вывода:
	Толстой — Война и мир (1425 стр.) — Не прочитана
	Оруэлл — 1984 (328 стр.) — Прочитана
	Решение:
	class Book:
	def __init__(self, title, author, pages):
	self.title = title
	self.author = author
	self.pages = pages
	self.is_read = False
	
	def read(self):
	self.is_read = True
	
	def info(self):
	status = "Прочитана" if self.is_read else "Не прочитана"
	print(f'{self.author} — {self.title} ({self.pages} стр.) — {status}')
	
	title1 = input()

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
author1 = input()
pages1 = int(input())
book1 = Book(title1, author1, pages1)

title2 = input()
author2 = input()
pages2 = int(input())
book2 = Book(title2, author2, pages2)
book2.read()

book1.info()
book2.info()
```

2. Класс «Товар» с методами класса и статическим методом

Создай класс Product с атрибутами: name (название) и price (цена – целое число). Реализуй:

конструктор __init__(name, price);

статический метод is_affordable(price, budget), который возвращает True, если price <= budget, и False в противном случае;

метод класса from_string(data), который принимает строку в формате «название: цена» (например, «Телефон: 50000»), создаёт и возвращает объект Product;

метод __str__, возвращающий строку «Товар: {name}, цена: {price} руб.».

В основной программе прочитай одну строку в формате «название: цена», создай объект с помощью from_string() и выведи его строковое представление. Затем прочитай бюджет (целое число) и выведи, доступен ли товар по цене.

Входные данные:

Первая строка – «название: цена».

Вторая строка – бюджет (целое число).

Выходные данные:

Строковое представление товара и True/False – доступность по цене.

Пример ввода:

Телефон: 50000

45000

Пример вывода:

Товар: Телефон, цена: 50000 руб.

False

Решение:

```
class Product:
    def __init__(self, name, price):
        self.name = name
        self.price = price

    @staticmethod
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
    def is_affordable(price, budget):
        return price <= budget

    @classmethod
    def from_string(cls, data):
        name, price = data.split(': ')
        return cls(name, int(price))

    def __str__(self):
        return f'Товар: {self.name}, цена: {self.price} руб.'

data = input()
budget = int(input())
product = Product.from_string(data)
print(product)
print(Product.is_affordable(product.price, budget))
```

3. Класс «Дробь» с сокращением и умножением

Создай класс Fraction для работы с обыкновенными дробями. Конструктор принимает числитель и знаменатель (целые числа). Реализуй:

метод __str__, возвращающий строку «числитель/знаменатель» (несократимую дробь);

метод __mul__ для умножения двух дробей, возвращающий новый объект Fraction (сокращённый);

статический метод gcd(a, b) для нахождения наибольшего общего делителя (используй алгоритм Евклида).

В основной программе прочитай две строки, каждая содержит два целых числа – числитель и знаменатель первой и второй дроби. Выведи произведение этих дробей.

Входные данные:

Две строки, в каждой – два целых числа через пробел.

Выходные данные:

Одна строка – произведение дробей в формате «числитель/знаменатель».

Пример ввода:

2 3

4 5

Пример вывода:

8/15

Решение:

```
class Fraction:
    def __init__(self, num, den):
        self.num = num
        self.den = den
        self._reduce()
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
    def _reduce(self):
        g = self.gcd(abs(self.num), abs(self.den))
        self.num //= g
        self.den //= g

    @staticmethod
    def gcd(a, b):
        while b:
            a, b = b, a % b
        return a

    def __mul__(self, other):
        new_num = self.num * other.num
        new_den = self.den * other.den
        return Fraction(new_num, new_den)

    def __str__(self):
        return f'{self.num}/{self.den}'

n1, d1 = map(int, input().split())
f1 = Fraction(n1, d1)
n2, d2 = map(int, input().split())
f2 = Fraction(n2, d2)

print(f1 * f2)
```

4. Класс «Студент» с переводом на следующий курс

Создай класс Student с атрибутами name (имя) и course (курс – целое число от 1 до 4). Реализуй:

конструктор __init__(name, course), который проверяет корректность курса: если курс не от 1 до 4, установи курс в 1 и выведи сообщение «Курс исправлен на 1»;

метод promote() – переводит студента на следующий курс, если текущий курс меньше 4; если курс равен 4, вывести «Студент выпустился» и не менять курс;

метод __str__, возвращающий строку «{name}, курс {course}».

В основной программе прочитай количество студентов n, затем для каждого студента – имя и курс (через пробел). Создай список студентов. Затем примени метод promote() к каждому студенту и выведи их строковое представление (по одному на строке).

Входные данные:

Первая строка – целое число n.

Далее n строк, каждая содержит имя и курс (целое число).

Выходные данные:

Сообщения об исправлении курса (если были) и после применения promote() – строки студентов.

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Пример ввода:
3
Анна 2
Борис 5
Вика 4
Пример вывода:
Курс исправлен на 1
Студент выпустился
Анна, курс 3
Борис, курс 2
Вика, курс 4
Решение:

```
class Student:
    def __init__(self, name, course):
        self.name = name
        if 1 <= course <= 4:
            self.course = course
        else:
            print('Курс исправлен на 1')
            self.course = 1

    def promote(self):
        if self.course == 4:
            print('Студент выпустился')
        else:
            self.course += 1

    def __str__(self):
        return f'{self.name}, курс {self.course}'

n = int(input())
students = []
for _ in range(n):
    name, course = input().split()
    students.append(Student(name, int(course)))

for s in students:
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

	s.promote()
	
	for s in students:
	print(s)
	5. Класс «Вектор» (сложение и сравнение длин)
	Создай класс Vector2D для двумерного вектора с координатами x и y (вещественные числа). Реализуй:
	конструктор __init__(x, y);
	метод __str__, возвращающий «Вектор (x, y)»;
	метод length(), возвращающий длину вектора (вещественное число);
	перегрузку операторов: + (покоординатное сложение), - (покоординатное вычитание);
	метод __eq__, сравнивающий длины векторов (равны, если длины равны);
	метод __lt__, сравнивающий длины векторов (True, если длина первого меньше длины второго).
	В основной программе прочитай координаты двух векторов (четыре числа через пробел: x1 y1 x2 y2). Создай два вектора, выведи их сумму и разность, а также результат сравнения длин (True/False).
	Входные данные:
	Четыре числа через пробел: x1 y1 x2 y2.
	Выходные данные:
	Три строки: сумма векторов, разность векторов, результат сравнения длин (<).
	Пример ввода:
	1 2 3 4
	Пример вывода:
	Вектор (4.0, 6.0)
	Вектор (-2.0, -2.0)
	True
	Решение:
	import math
	
	class Vector2D:
	def __init__(self, x, y):
	self.x = x
	self.y = y
	
	def length(self):
	return math.hypot(self.x, self.y)
	
	def __str__(self):
	return f'Вектор ({self.x}, {self.y})'

	def __add__(self, other):
	return Vector2D(self.x + other.x, self.y + other.y)
	
	def __sub__(self, other):
	return Vector2D(self.x - other.x, self.y - other.y)
	
	def __eq__(self, other):
	return self.length() == other.length()
	
	def __lt__(self, other):
	return self.length() < other.length()
	
	x1, y1, x2, y2 = map(float, input().split())
	v1 = Vector2D(x1, y1)
	v2 = Vector2D(x2, y2)
	
	print(v1 + v2)
	print(v1 - v2)
	print(v1 < v2)

	Модуль 2, Объектно-ориентированное программирование (ООП), Парадигмы ООП
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи
Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода

Контур Крипто
владелец Маршеня Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3191c94f2cb04d6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполняет не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)

Варианты заданий	1. Наследование: транспортные средства
	Создай базовый класс Vehicle с атрибутом name (название) и методом start_engine(), который возвращает строку «Двигатель запущен».
	Создай два дочерних класса:
	Car – переопределяет метод start_engine(), возвращая «Автомобиль {name}: двигатель запущен».
	Motorcycle – переопределяет метод start_engine(), возвращая «Мотоцикл {name}: двигатель запущен».
	В основной программе прочитай две строки:
	первая – тип транспорта ("Car" или "Motorcycle");
	вторая – название (строка).
	Создай объект соответствующего класса и выведи результат вызова start_engine().
	Входные данные:
	Две строки: тип и название.
	Выходные данные:
	Одна строка – результат работы метода start_engine().
	Пример ввода:
	Car
	Tesla
	Пример вывода:
	Автомобиль Tesla: двигатель запущен
	Решение:
	class Vehicle:
	  def __init__(self, name):
	  self.name = name
	  def start_engine(self):
	  return 'Двигатель запущен'
	
	class Car(Vehicle):
	  def start_engine(self):
	  return f'Автомобиль {self.name}: двигатель запущен'
	
	class Motorcycle(Vehicle):
	  def start_engine(self):
	  return f'Мотоцикл {self.name}: двигатель запущен'
	
	t = input()
	name = input()
	if t == 'Car':
	  v = Car(name)

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
else:
    v = Motorcycle(name)
print(v.start_engine())
```

2. Множественное наследование: работник и студент

Создай класс Worker с методом work(), возвращающим строку «Работаю над проектом».
Создай класс Student с методом study(), возвращающим строку «Изучаю Python».
Создай класс WorkingStudent, наследующий от Worker и Student (в этом порядке).
В основной программе создай объект класса WorkingStudent и выведи результаты вызова методов work() и study() (каждый на отдельной строке).
Входные данные:
Ничего не вводится.
Выходные данные:
Две строки.
Пример вывода:
Работаю над проектом
Изучаю Python
Решение:

```
class Worker:
    def work(self):
        return 'Работаю над проектом'

class Student:
    def study(self):
        return 'Изучаю Python'

class WorkingStudent(Worker, Student):
    pass

ws = WorkingStudent()
print(ws.work())
print(ws.study())
```

3. Полиморфизм: объём фигур (цилиндр и прямоугольный параллелепипед)

Создай класс Cylinder (цилиндр) с атрибутами radius и height (вещественные числа).
Реализуй метод volume(), возвращающий объём цилиндра по формуле $\pi * r^2 * h$ (используй math.pi).
Создай класс Box (прямоугольный параллелепипед) с атрибутами length, width, height.
Реализуй метод volume(), возвращающий объём length * width * height.
В основной программе прочитай две строки:
первая – тип фигуры ("Cylinder" или "Box");

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

вторая – три числа через пробел (для цилиндра: радиус и высота; для параллелепипеда: длина, ширина, высота).
Создай объект соответствующего класса и выведи его объём, округлённый до двух знаков после запятой.
Входные данные:
Две строки: тип фигуры и три числа через пробел.
Выходные данные:
Одно вещественное число с двумя знаками после запятой.
Пример ввода:
Cylinder
3 5
Пример вывода:
141.37
Решение:

```
import math

class Cylinder:
    def __init__(self, radius, height):
        self.radius = radius
        self.height = height
    def volume(self):
        return math.pi * self.radius ** 2 * self.height

class Box:
    def __init__(self, length, width, height):
        self.length = length
        self.width = width
        self.height = height
    def volume(self):
        return self.length * self.width * self.height

t = input()
vals = list(map(float, input().split()))
if t == 'Cylinder':
    obj = Cylinder(vals[0], vals[1])
else:
    obj = Box(vals[0], vals[1], vals[2])
print(round(obj.volume(), 2))
```

4. Инкапсуляция: температура с проверкой на абсолютный ноль

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Создай класс Temperature с приватным атрибутом __celsius.
Реализуй:
конструктор __init__(celsius), который устанавливает начальную температуру, но если значение меньше -273.15, выводит «Недопустимая температура, установлено 0» и устанавливает 0.
свойство celsius (геттер), возвращающее значение.
сеттер celsius, который позволяет изменить значение, только если оно >= -273.15; иначе выводит «Температура не может быть ниже абсолютного нуля» и не меняет значение.
метод __str__, возвращающий строку «Температура: {значение} °C».
В основной программе прочитай два числа (начальная температура и новая температура). Создай объект с начальной температурой, затем попробуй установить новую температуру через сеттер, выведи строковое представление объекта.
Входные данные:
Два целых числа (начальная и новая температура) через пробел.
Выходные данные:
Сообщения об ошибках (если были) и строковое представление.
Пример ввода:
-300 25
Пример вывода:
Недопустимая температура, установлено 0
Температура: 25 °C
Решение:

```
class Temperature:
    def __init__(self, celsius):
        self.__celsius = 0
        self.celsius = celsius

    @property
    def celsius(self):
        return self.__celsius

    @celsius.setter
    def celsius(self, value):
        if value < -273.15:
            print('Недопустимая температура, установлено 0')
            self.__celsius = 0
        else:
            self.__celsius = value

    def __str__(self):
        return f'Температура: {self.__celsius} °C'
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

start, new = map(int, input().split())
t = Temperature(start)
t.celsius = new
print(t)
5. Инкапсуляция: банковский счёт с приватным балансом
Создай класс BankAccount с приватным атрибутом __balance (начальное значение 0).
Реализуй:
метод deposit(amount) – увеличивает баланс на amount, если amount > 0, иначе выводит «Сумма должна быть положительной»;
метод withdraw(amount) – уменьшает баланс на amount, если amount > 0 и amount <= __balance, иначе выводит соответствующее сообщение («Недостаточно средств» или «Сумма должна быть положительной»);
геттер balance через @property, возвращающий текущий баланс.
В основной программе прочитай количество операций n, затем n строк команд вида:
deposit X или withdraw X или balance (вывести текущий баланс).
Выведи результаты вызовов balance и сообщения об ошибках (если они возникают).
Гарантируется, что команда balance может встретиться несколько раз.
Входные данные:
Первая строка – целое число n.
Далее n строк команд.
Выходные данные:
Результаты выполнения команд balance и сообщения об ошибках – каждый результат на новой строке.
Пример ввода:
4
deposit 500
withdraw 200
balance
withdraw 400
Пример вывода:
Баланс: 300
Недостаточно средств
Решение:

```
class BankAccount:
    def __init__(self):
        self.__balance = 0

    @property
    def balance(self):
        return self.__balance
    def deposit(self, amount):
        if amount <= 0:
            print('Сумма должна быть положительной')
        else:
            self.__balance += amount

    def withdraw(self, amount):
        if amount <= 0:
            print('Сумма должна быть положительной')
        elif amount > self.__balance:
            print('Недостаточно средств')
        else:
            self.__balance -= amount

acc = BankAccount()
n = int(input())
for _ in range(n):
    cmd = input().split()
    if cmd[0] == 'deposit':
        acc.deposit(int(cmd[1]))
    elif cmd[0] == 'withdraw':
        acc.withdraw(int(cmd[1]))
    elif cmd[0] == 'balance':
        print(f'Баланс: {acc.balance}')
```

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Модуль 2, Объектно-ориентированное программирование (ООП), Командный проект	
Количество академических часов	1

Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи
Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполняет не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)

Варианты заданий	1. Наследование: базовый класс «Сотрудник» и дочерний «Менеджер»
	Создай базовый класс Employee с атрибутами name (имя) и salary (оклад – целое число). Реализуй:
	конструктор __init__(name, salary);
	метод get_info(), возвращающий строку «Сотрудник: {name}, оклад: {salary} руб.».
	Создай класс Manager, наследующий от Employee. Добавь атрибут bonus (премия – целое число). Переопредели метод get_info() так, чтобы он возвращал строку:
	«Менеджер: {name}, оклад: {salary} + премия {bonus} = {salary+bonus} руб.».
	В основной программе прочитай две строки: для сотрудника (имя и оклад через пробел) и для менеджера (имя, оклад, премия через пробел). Создай объекты соответствующих классов и выведи результат вызова get_info() для каждого (сначала для сотрудника, затем для менеджера).
	Входные данные:
	Две строки:
	<имя_сотрудника> <оклад>
	<имя_менеджера> <оклад> <премия>
	Выходные данные:
	Две строки – результаты get_info().
	Пример ввода:
	Смирнова 50000
	Романов 80000 20000
	Пример вывода:
	Сотрудник: Смирнова, оклад: 50000 руб.
	Менеджер: Романов, оклад: 80000 + премия 20000 = 100000 руб.
	Решение:
	class Employee:
	def __init__(self, name, salary):
	self.name = name
	self.salary = salary
	
	def get_info(self):
	return f'Сотрудник: {self.name}, оклад: {self.salary} руб.'
	
	class Manager(Employee):
	def __init__(self, name, salary, bonus):
	super().__init__(name, salary)
	self.bonus = bonus
	
	def get_info(self):
	total = self.salary + self.bonus
	return f'Менеджер: {self.name}, оклад: {self.salary} + премия {self.bonus} = {total} руб.'

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
emp_name, emp_salary = input().split()
emp = Employee(emp_name, int(emp_salary))

mgr_name, mgr_salary, mgr_bonus = input().split()
mgr = Manager(mgr_name, int(mgr_salary), int(mgr_bonus))

print(emp.get_info())
print(mgr.get_info())
```

2. Множественное наследование: «Триатлонист»

Создай класс Swimmer с методом swim(), возвращающим строку «Плыву».
Создай класс Cyclist с методом cycle(), возвращающим строку «Еду на велосипеде».
Создай класс Runner с методом run(), возвращающим строку «Бегу».
Создай класс Triathlete, наследующий от Swimmer, Cyclist и Runner (в этом порядке). В классе Triathlete переопредели метод swim() так, чтобы он возвращал строку «Плыву в открытой воде».
В основной программе создай объект класса Triathlete и выведи результаты вызова всех трёх методов: swim(), cycle(), run() (каждый на отдельной строке).
Входные данные:
Ничего не вводится.
Выходные данные:
Три строки.
Пример вывода:
Плыву в открытой воде
Еду на велосипеде
Бегу
Решение:

```
class Swimmer:
    def swim(self):
        return 'Плыву'

class Cyclist:
    def cycle(self):
        return 'Еду на велосипеде'

class Runner:
    def run(self):
        return 'Бегу'

class Triathlete(Swimmer, Cyclist, Runner):
    def swim(self):
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

```
        return 'Плыву в открытой воде'

t = Triathlete()
print(t.swim())
print(t.cycle())
print(t.run())
```

3. Полиморфизм: объём фигур (цилиндр и параллелепипед)

Создай класс Cylinder (цилиндр) с атрибутами radius и height (вещественные числа). Реализуй метод volume(), возвращающий объём цилиндра по формуле $\pi * r^2 * h$ (используй math.pi).

Создай класс Box (прямоугольный параллелепипед) с атрибутами length, width, height. Реализуй метод volume(), возвращающий объём length * width * height.

В основной программе прочитай две строки:

первая – тип фигуры ("Cylinder" или "Box");

вторая – три числа через пробел (для цилиндра: радиус, высота; для параллелепипеда: длина, ширина, высота).

Создай объект соответствующего класса и выведи его объём, округлённый до двух знаков после запятой.

Входные данные:

Две строки: тип фигуры и три числа через пробел.

Выходные данные:

Одно вещественное число с двумя знаками после запятой.

Пример ввода:

```
Cylinder
3 5
```

Пример вывода:

```
141.37
```

Решение:

```
import math

class Cylinder:
    def __init__(self, radius, height):
        self.radius = radius
        self.height = height
    def volume(self):
        return math.pi * self.radius ** 2 * self.height

class Box:
    def __init__(self, length, width, height):
        self.length = length
        self.width = width
        self.height = height
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
    def volume(self):
        return self.length * self.width * self.height

t = input()
vals = list(map(float, input().split()))
if t == 'Cylinder':
    obj = Cylinder(vals[0], vals[1])
else:
    obj = Box(vals[0], vals[1], vals[2])
print(round(obj.volume(), 2))
```

4. Инкапсуляция: класс «Температура» с проверкой на абсолютный ноль

Создай класс Temperature с приватным атрибутом __celsius. Реализуй:

конструктор __init__(celsius), который устанавливает начальную температуру, но если значение меньше -273.15, выводит «Недопустимая температура, установлено 0» и устанавливает 0.

геттер celsius через @property, возвращающий значение.

сеттер celsius, который позволяет изменить значение, только если оно >= -273.15; иначе выводит «Температура не может быть ниже абсолютного нуля» и не меняет значение.

метод __str__, возвращающий строку «Температура: {значение} °C».

В основной программе прочитай два числа (начальная температура и новая температура). Создай объект с начальной температурой, затем попробуй установить новую температуру через сеттер, выведи строковое представление объекта.

Входные данные:

Два целых числа (начальная и новая температура) через пробел.

Выходные данные:

Сообщения об ошибках (если были) и строковое представление.

Пример ввода:

-300 25

Пример вывода:

Недопустимая температура, установлено 0
Температура: 25 °C

Решение:

```
class Temperature:
    def __init__(self, celsius):
        self.__celsius = 0
        self.celsius = celsius

    @property
    def celsius(self):
        return self.__celsius

    @celsius.setter
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
    def celsius(self, value):
        if value < -273.15:
            print('Недопустимая температура, установлено 0')
            self.__celsius = 0
        else:
            self.__celsius = value

    def __str__(self):
        return f'Температура: {self.__celsius} °C'

start, new = map(int, input().split())
t = Temperature(start)
t.celsius = new
print(t)
```

5. Класс «Пароль» с собственными исключениями

Создай класс Password. Конструктор принимает строку с паролем и выполняет проверку:
длина пароля не менее 8 символов (иначе вызывается исключение MinLengthError);
пароль должен содержать хотя бы один символ из набора !@#$%^&*()<>?, (иначе вызывается исключение NoSpecialCharError).
Если пароль корректен, он сохраняется в атрибут __password. Если нет – программа выводит соответствующее сообщение («Короткий пароль» или «Нет специального символа»).
Создай собственные классы исключений, унаследованные от Exception.
В основной программе прочитай одну строку – пароль. Создай объект класса Password с этим паролем. Если пароль корректен, выведи «Пароль: {пароль}».
Входные данные:
Одна строка – пароль.
Выходные данные:
Сообщение об ошибке или «Пароль: ...».
Пример ввода:
12345678
Пример вывода:
Нет специального символа
Пример ввода 2:
12345678@
Пример вывода 2:
Пароль: 12345678@
Решение:

```
class MinLengthError(Exception):
    pass

class NoSpecialCharError(Exception):
    pass

class Password:
    SPECIAL = '!@#$%^&*()<>?,'

    def __init__(self, password):
        if len(password) < 8:
            raise MinLengthError('Короткий пароль')
        if not any(ch in self.SPECIAL for ch in password):
            raise NoSpecialCharError('Нет специального символа')
        self.__password = password

    def __str__(self):
        return f'Пароль: {self.__password}'

s = input()
try:
    p = Password(s)
    print(p)
except MinLengthError as e:
    print(e)
except NoSpecialCharError as e:
    print(e)
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Модуль 3, Алгоритмы и структуры данных, Структуры данных

Количество академических часов	1

Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи
Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполняет не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)

Варианты заданий	1. Поиск элемента в связном списке
	Напиши программу, которая находит номер позиции (нумерация с 1) заданного значения в связном списке. Если значение не найдено, программа должна вывести -1. Связный список содержит следующие значения: pineapple -> mango -> pear -> cherry -> None.
	Входные данные:
	Одна строка – значение для поиска.
	Выходные данные:
	Одно целое число – номер позиции или -1.
	Пример ввода:
	mango
	Пример вывода:
	2
	Пример ввода 2:
	grape
	Пример вывода 2:
	-1
	Решение:
	class Node:
	  def __init__(self, value, next):
	  self.value = value
	  self.next = next
	
	n4 = Node('cherry', None)
	n3 = Node('pear', n4)
	n2 = Node('mango', n3)
	n1 = Node('pineapple', n2)
	
	s = input()
	node = n1
	index = 1
	while node is not None:
	  if node.value == s:
	  print(index)
	  break
	  node = node.next
	  index += 1
	else:
	  print(-1)
	2. Удаление элемента из связного списка

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Напиши программу для удаления элемента из связного списка по его номеру (нумерация с 0). Гарантируется, что будет введён номер существующего элемента. После удаления выведи элементы списка через функцию print_linked_list(). Список содержит значения: red -> green -> blue -> yellow -> None.
Входные данные:
Одно целое число – номер элемента для удаления.
Выходные данные:
Строка с элементами списка после удаления.
Пример ввода:
1
Пример вывода:
red -> blue -> yellow -> None
Решение:

```
class Node:
    def __init__(self, value, next):
        self.value = value
        self.next = next

    def __str__(self):
        return self.value

def print_linked_list(vertex):
    while vertex:
        print(vertex, end=" -> ")
        vertex = vertex.next
    print("None")

def delete_node(head, index):
    if index == 0:
        return head.next
    node = head
    i = 0
    while i < index - 1:
        node = node.next
        i += 1
    node.next = node.next.next
    return head

n4 = Node('yellow', None)
n3 = Node('blue', n4)
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
n2 = Node('green', n3)
n1 = Node('red', n2)

n = int(input())
head = delete_node(n1, n)
print_linked_list(head)
```

3. Проверка правильной скобочной последовательности (только круглые скобки)

Напиши программу, которая проверяет, является ли введённая строка правильной скобочной последовательностью. Строка состоит только из символов (и).
Для проверки используй стек. Программа должна вывести True, если последовательность правильная, и False – если нет.
Входные данные:
Одна строка, состоящая из символов (и).
Выходные данные:
True или False.
Пример ввода:
(()()(()))
Пример вывода:
True
Пример ввода 2:
())(()
Пример вывода 2:
False
Решение:

```
s = input()
stack = []
for char in s:
    if char == '(':
        stack.append(char)
    elif char == ')':
        if not stack:
            print(False)
            break
        stack.pop()
else:
    if stack:
        print(False)
    else:
        print(True)
```

4. Вычисление выражения в обратной польской записи

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Напиши программу, которая вычисляет значение арифметического выражения, записанного в обратной польской записи (операнды и операторы разделены пробелами). Операции: +, -, *, / (целочисленное деление). Используй стек для вычислений.
Входные данные:
Строка, содержащая числа и операции, разделённые пробелами.
Выходные данные:
Одно целое число – результат вычисления.
Пример ввода:
5 3 2 * +
Пример вывода:
11
Решение:

```
class Stack:
    def __init__(self):
        self.items = []

    def push(self, value):
        self.items.append(value)

    def pop(self):
        return self.items.pop()

stack = Stack()
line = input()
ops = '+-*/'
for elem in line.split():
    if elem in ops:
        b = stack.pop()
        a = stack.pop()
        if elem == '+':
            stack.push(a + b)
        elif elem == '-':
            stack.push(a - b)
        elif elem == '*':
            stack.push(a * b)
        elif elem == '/':
            stack.push(a // b)
    else:
        stack.push(int(elem))
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

	print(stack.pop())
	5. Класс Deque (дек) с методами push_front, push_back, pop_front, pop_back, size
	Создай класс Deque с методами:
	push_back(item) – добавляет элемент в конец дека.
	pop_back() – выводит и удаляет последний элемент; если дек пуст, выводит 'error'.
	push_front(item) – добавляет элемент в начало дека.
	pop_front() – выводит и удаляет первый элемент; если дек пуст, выводит 'error'.
	size() – выводит текущий размер дека.
	Для проверки используется код eval(input()). Твой класс должен корректно работать с приведёнными командами.
	Входные данные:
	Выражение (например, [deque := Deque(), deque.push_back(1), deque.pop_front(), deque.size()]).
	Выходные данные:
	Результаты вызовов pop_back(), pop_front(), size().
	Пример ввода:
	[deque := Deque(), deque.push_front(5), deque.push_front(7), deque.pop_front(), deque.pop_front(), deque.pop_front()]
	Пример вывода:
	7
	5
	error
	Решение:
	class Deque:
	  def __init__(self):
	  self.items = []
	
	  def push_back(self, item):
	  self.items.append(item)
	
	  def push_front(self, item):
	  self.items.insert(0, item)
	
	  def pop_back(self):
	  if self.items:
	  print(self.items.pop())
	  else:
	  print('error')
	
	  def pop_front(self):

	if self.items:
	  print(self.items.pop(0))
	  else:
	  print('error')
	
	  def size(self):
	  print(len(self.items))
	
	eval(input())

Модуль 3, Алгоритмы и структуры данных, Алгоритмы поиска и сортировки	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи
Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполняет не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)

Контур Крипто
владелец Маршеня Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Варианты заданий	Задача 1. Сортировка «тяжёлым камнем» (по убыванию)
	Напиши программу, которая сортирует список целых чисел по убыванию (от большего к меньшему) с помощью алгоритма «тяжёлый камень» (аналог пузырька, но самый «тяжёлый» элемент перемещается в начало). После каждого внешнего прохода (т.е. после того, как один элемент встал на своё место) выводи текущее состояние списка. Всего должно быть выведено n-1 строка (где n – размер списка).
	Входные данные
	Первая строка – целое число n (размер списка).
	Вторая строка – n целых чисел, разделённых пробелами.
	Выходные данные
	n-1 строка, в каждой – элементы списка через пробел в текущем порядке.
	Пример
	Ввод:
	5
	7 2 9 1 5
	Вывод:
	9 2 7 1 5
	9 7 2 1 5
	9 7 5 1 2
	9 7 5 2 1
	Примечание
	Подсказки внутри input() не используй.
	Решение
	n = int(input())
	arr = list(map(int, input().split()))
	for i in range(n - 1):
	  for j in range(n - 2, i - 1, -1):
	  if arr[j] < arr[j + 1]:
	  arr[j], arr[j + 1] = arr[j + 1], arr[j]
	  print(' '.join(map(str, arr)))
	Задача 2. Сортировка выбором максимума (по возрастанию)
	Реализуй сортировку списка целых чисел по возрастанию с помощью алгоритма сортировки выбором (на каждом шаге ищется максимальный элемент и ставится в конец неотсортированной части). После каждого внешнего прохода выводи текущее состояние списка. Вывести n-1 строку.
	Входные данные
	Первая строка – целое число n.
	Вторая строка – n целых чисел через пробел.
	Выходные данные
	n-1 строка с элементами списка после каждого прохода.
	Пример
	Ввод:
	5
	7 2 9 1 5

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Вывод:
7 2 5 1 9
1 2 5 7 9
1 2 5 7 9
1 2 5 7 9
Примечание
Подсказки внутри input() не используй.
Решение

```
n = int(input())
arr = list(map(int, input().split()))
for i in range(n - 1, 0, -1):
    max_idx = 0
    for j in range(1, i + 1):
        if arr[j] > arr[max_idx]:
            max_idx = j
    arr[i], arr[max_idx] = arr[max_idx], arr[i]
    print(' '.join(map(str, arr)))
```

Задача 3. Составление наибольшего числа из цифр
Дана строка, содержащая несколько целых чисел (каждое может быть многозначным). Требуется переставить эти числа в таком порядке, чтобы составленное из них подряд одно большое число было максимально возможным. Выведи это число.
Используй сортировку вставками с кастомной функцией сравнения: два числа a и b считаются расположенными в правильном порядке, если str(a) + str(b) >= str(b) + str(a).
Входные данные
Строка целых чисел через пробел.
Выходные данные
Одно целое число (без разделителей).
Пример
Ввод:
15 56 3
Вывод:
56315
Примечание
Подсказки внутри input() не используй.
Решение

```
def greater(a, b):
    return str(a) + str(b) > str(b) + str(a)

nums = list(map(int, input().split()))
# Сортировка вставками
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
for i in range(1, len(nums)):
    key = nums[i]
    j = i - 1
    while j >= 0 and greater(key, nums[j]):
        nums[j + 1] = nums[j]
        j -= 1
    nums[j + 1] = key
print(''.join(map(str, nums)))
```

Задача 4. Левый двоичный поиск (первое вхождение)

Дан список целых чисел, отсортированный по неубыванию, и число x. Реализуй левый двоичный поиск – найди индекс первого элемента, равного x. Если таких элементов нет, выведи -1. Используй алгоритм двоичного поиска (не встроенные методы index).

Входные данные

Первая строка – целое число n (размер списка).

Вторая строка – n целых чисел через пробел (гарантированно отсортированы по неубыванию).

Третья строка – целое число x.

Выходные данные

Одно целое число – индекс первого вхождения x (индексация с 0) или -1.

Пример

Ввод:

```
8
1 3 3 3 7 9 9 12
3
```

Вывод:

```
1
```

Примечание

Подсказки внутри input() не используй.

Решение

```
n = int(input())
arr = list(map(int, input().split()))
x = int(input())
left, right = 0, n
found = -1
while left < right:
    mid = (left + right) // 2
    if arr[mid] < x:
        left = mid + 1
    elif arr[mid] > x:
        right = mid
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

	else:
	if mid == 0 or arr[mid - 1] != x:
	found = mid
	break
	else:
	right = mid
	print(found)
	Задача 5. Правый двоичный поиск (последнее вхождение)
	Дан список целых чисел, отсортированный по неубыванию, и число x. Реализуй правый двоичный поиск – найди индекс последнего элемента, равного x. Если таких элементов нет, выведи -1.
	Входные данные
	Первая строка – целое число n.
	Вторая строка – n целых чисел через пробел (отсортированы по неубыванию).
	Третья строка – целое число x.
	Выходные данные
	Одно целое число – индекс последнего вхождения x или -1.
	Пример
	Ввод:
	8
	1 3 3 3 7 9 9 12
	3
	Вывод:
	3
	Примечание
	Подсказки внутри input() не используй.
	Решение
	n = int(input())
	arr = list(map(int, input().split()))
	x = int(input())
	left, right = 0, n
	found = -1
	while left < right:
	mid = (left + right) // 2
	if arr[mid] < x:
	left = mid + 1
	elif arr[mid] > x:
	right = mid
	else:

	if mid == n - 1 or arr[mid + 1] != x:
	found = mid
	break
	else:
	left = mid + 1
	print(found)

Модуль 3, Алгоритмы и структуры данных, Ответственное использование ИИ-инструментов	
Количество академических часов	
Формы контроля	
Диагностические инструменты	
Показатели и критерии оценивания	
Модуль 3, Алгоритмы и структуры данных, Сложность алгоритмов	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи
Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполняет не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Маринова Виктория Вадимовна
серийный номер 01b83191c941e2024c0127cc0f3aeaad
срок действия 05.12.2025 - 05.03.2027

Варианты заданий	1. Класс Queue (очередь) на связном списке
	Создай класс Queue, реализующий очередь на односвязном списке с операциями:
	push(item) – добавляет элемент в конец очереди (O(1)).
	pop() – выводит и удаляет первый элемент; если очередь пуста, выводит 'error'.
	size() – выводит текущий размер очереди.
	В первой строке вводится число команд n. В следующих n строках – команды: push x, pop, size. Выполни команды и выведи результаты.
	Входные данные:
	Целое число n, затем n строк с командами.
	Выходные данные:
	Результаты выполнения команд pop и size.
	Пример ввода:
	5
	push 10
	push 20
	size
	pop
	pop
	Пример вывода:
	2
	10
	error
	Решение:
	class Node:
	def __init__(self, value, next_node):
	self.value = value
	self.next = next_node
	
	class Queue:
	def __init__(self):
	self.head = None
	self.tail = None
	self.size_count = 0
	
	def push(self, x):
	if not self.head:
	self.head = Node(x, None)
	self.tail = self.head

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
        else:
            new_node = Node(x, None)
            self.tail.next = new_node
            self.tail = new_node
        self.size_count += 1

    def pop(self):
        if self.size_count == 0:
            print('error')
        else:
            print(self.head.value)
            self.head = self.head.next
            self.size_count -= 1

    def size(self):
        print(self.size_count)

n = int(input())
q = Queue()
for _ in range(n):
    cmd = input().split()
    if cmd[0] == 'push':
        q.push(int(cmd[1]))
    elif cmd[0] == 'pop':
        q.pop()
    elif cmd[0] == 'size':
        q.size()
```

2. Циклический сдвиг дека вправо

Напиши программу, которая выполняет циклический сдвиг элементов дека на k позиций вправо. При сдвиге последний элемент перемещается в начало. Реализуй класс Deque с методами push_back, pop_back, push_front, size. Для сдвига используй методы дека, не преобразуя его в список.

Входные данные:

В первой строке – элементы дека через пробел. Во второй строке – целое число k.

Выходные данные:

Элементы дека после сдвига через пробел.

Пример ввода:

10 20 30 40 50

3

Пример вывода:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

30 40 50 10 20
Решение:

```
class Deque:
    def __init__(self):
        self.items = []

    def push_back(self, item):
        self.items.append(item)

    def pop_back(self):
        if self.items:
            return self.items.pop()
        return None

    def push_front(self, item):
        self.items.insert(0, item)

    def size(self):
        return len(self.items)

    def print_deque(self):
        print(*self.items)

data = list(map(int, input().split()))
k = int(input())
d = Deque()
for val in data:
    d.push_back(val)

if d.size() > 0:
    k = k % d.size()
    for _ in range(k):
        last = d.pop_back()
        d.push_front(last)

d.print_deque()
```

3. Проверка правильной скобочной последовательности (несколько типов)

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Напиши программу, которая проверяет, является ли введённая строка правильной скобочной последовательностью. Поддерживаются три типа скобок: (), [], {}.
Используй стек для проверки.
Входные данные:
Одна строка, содержащая только символы ()[]{}.
Выходные данные:
True, если последовательность правильная, иначе False.
Пример ввода:
({[]})
Пример вывода:
True
Пример ввода 2:
({[}])
Пример вывода 2:
False
Решение:

```
s = input()
stack = []
pairs = {')': '(', ']': '[', '}': '{'}
for ch in s:
    if ch in '([{':
        stack.append(ch)
    else:
        if not stack or stack[-1] != pairs[ch]:
            print(False)
            break
        stack.pop()
else:
    print(len(stack) == 0)
```

4. Поиск пары чисел с заданной суммой в отсортированном массиве
Напиши программу, которая определяет, есть ли в отсортированном по неубыванию массиве два различных элемента, сумма которых равна заданному числу k.
Используй алгоритм с двумя указателями (O(n)). Программа должна вывести True или False.
Входные данные:
В первой строке – размер массива n. Во второй строке – n целых чисел через пробел (отсортированы по неубыванию). В третьей строке – число k.
Выходные данные:
True или False.
Пример ввода:
5
1 3 5 7 9
12

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

	Пример вывода:
	True
	Решение:
	n = int(input())
	arr = list(map(int, input().split()))
	k = int(input())
	left, right = 0, n - 1
	found = False
	while left < right:
	  s = arr[left] + arr[right]
	  if s == k:
	  found = True
	  break
	  elif s < k:
	  left += 1
	  else:
	  right -= 1
	print(found)
	5. Проверка подпоследовательности (два указателя)
	Напиши программу, которая проверяет, является ли первая строка подпоследовательностью второй строки. Используй алгоритм с двумя указателями (O(n)).
	Подпоследовательность – строка, которую можно получить из другой удалением некоторых символов без изменения порядка оставшихся. Программа должна вывести True или False.
	Входные данные:
	Две строки (каждая на отдельной строке).
	Выходные данные:
	True, если первая строка является подпоследовательностью второй, иначе False.
	Пример ввода:
	abc
	axbdyc
	Пример вывода:
	False
	Пример ввода 2:
	ace
	abcde
	Пример вывода 2:
	True
	Решение:
	s1 = input().strip()
	s2 = input().strip()

	i = 0
	for ch in s2:
	  if i < len(s1) and ch == s1[i]:
	  i += 1
	print(i == len(s1))

Модуль 3, Алгоритмы и структуры данных, Командный проект	
Количество академических часов	
Формы контроля	
Диагностические инструменты	
Показатели и критерии оценивания	
Модуль 4, WEB и API, Работа с файлами	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи
Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполняет не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Мариона Виктория Вадимовна
серийный номер 01ab5101c94fe2024e0127ee0 заева
срок действия 05.12.2025 - 05.03.2027

Варианты заданий	1. Сумма чисел из файла
	Скачай файл data.txt, который содержит произвольное количество строк, каждая строка содержит целые числа, разделённые пробелом. Напиши программу, которая вычисляет сумму всех чисел из файла и выводит её.
	Входные данные:
	Файл data.txt (содержит числа).
	Выходные данные:
	Одно целое число – сумма всех чисел.
	Пример файла data.txt:
	10 20 30
	40 50
	60
	70 80
	90
	Пример вывода:
	450
	Решение:
	with open('data.txt') as file:
	total = 0
	for line in file:
	for num in line.split():
	total += int(num)
	print(total)
	2. Поиск самого высокого ученика в CSV
	Скачай файл students.csv, который содержит данные об учениках в формате CSV с заголовками: name, height (рост в сантиметрах). Напиши программу, которая находит имя ученика с наибольшим ростом и выводит его. Если таких несколько, выведи первого в порядке следования в файле.
	Входные данные:
	Файл students.csv.
	Выходные данные:
	Одна строка – имя ученика.
	Пример файла students.csv:
	name,height
	Анна,165
	Иван,180
	Мария,175
	Пётр,190
	Пример вывода:
	Пётр
	Решение:
	import csv

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
max_height = -1
max_name = ''
with open('students.csv', newline='', encoding='utf-8') as file:
    reader = csv.DictReader(file)
    for row in reader:
        h = int(row['height'])
        if h > max_height:
            max_height = h
            max_name = row['name']
print(max_name)
```

3. Создание CSV из JSON с курсами валют

Скачай файл currencies.json с информацией о курсах валют (структура аналогична rate.json). Напиши программу, которая создаёт CSV-файл currencies.csv с колонками Code и Name, записывает в него данные из JSON, сортируя по коду валюты (Code) в алфавитном порядке. Затем определи, на какой строке (нумерация с 1, включая заголовок) находится валюта с кодом EUR. Выведи это число.

Входные данные:

Файл currencies.json.

Выходные данные:

Одно целое число – номер строки, где находится EUR.

Пример (фрагмент currencies.json):

```
{
  "Valute": {
    "USD": {"CharCode": "USD", "Name": "Доллар США"},
    "EUR": {"CharCode": "EUR", "Name": "Евро"},
    "GBP": {"CharCode": "GBP", "Name": "Фунт стерлингов"}
  }
}
```

Пример currencies.csv:

```
Code,Name
EUR,Евро
GBP,Фунт стерлингов
USD,Доллар США
```

Пример вывода:

(для данного примера EUR на строке 2) – 2

Решение:

```
import json
import csv

with open('currencies.json', encoding='utf-8') as f:
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
    data = json.load(f)

rates = []
for code, info in data['Valute'].items():
    rates.append([info['CharCode'], info['Name']])
rates.sort(key=lambda x: x[0])

with open('currencies.csv', 'w', encoding='utf-8', newline='') as f:
    writer = csv.writer(f)
    writer.writerow(['Code', 'Name'])
    writer.writerows(rates)

# Поиск строки с EUR
for i, row in enumerate(rates, start=2):  # строка 1 - заголовок
    if row[0] == 'EUR':
        print(i)
        break
```

4. Группировка слов по последней букве (из стандартного ввода)

Напиши программу, которая читает произвольное количество строк из стандартного ввода (каждая строка содержит слова, разделённые пробелом), группирует слова по последней букве и выводит группы в алфавитном порядке букв. Внутри каждой группы слова выводятся в алфавитном порядке через пробел. Формат вывода: буква: слово1 слово2 .... Если на какую-то букву нет слов, она не выводится. Слова считаются состоящими только из букв (регистр сохраняется).

Входные данные:

Произвольное количество строк со словами.

Выходные данные:

Строки в формате буква: слова.

Пример ввода:

кот дом мост
стол лампа
парк торт

Пример вывода:

к: парк
м: дом лампа
п: лампа
т: кот мост стол торт

(пояснение: слова: кот(т), дом(м), мост(т), стол(л), лампа(а), парк(к), торт(т) – группировка по последней букве)

Решение:

```
from sys import stdin

groups = {}
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

	for line in stdin:
	  for word in line.split():
	  if word: # если не пустое
	  last = word[-1]
	  groups.setdefault(last, []).append(word)
	
	for letter in sorted(groups):
	  print(f'{letter}: {" ".join(sorted(groups[letter]))}')
	5. Калькулятор с аргументами командной строки (argparse)
	Напиши программу, которая принимает два целых числа как позиционные аргументы и опциональный ключ -op с выбором операции: add, sub, mul, div (по умолчанию add). Также добавь необязательный ключ -round для округления результата до целого (если указан, результат выводится как целое число, иначе как float). Программа должна вывести результат. При делении на ноль или неверной операции выводить Error.
	Примеры работы программы:
	$ python calc.py 5 3
	8
	$ python calc.py 5 3 -op mul
	15
	$ python calc.py 5 3 -op div -round
	2
	$ python calc.py 5 0 -op div
	Error
	Входные данные:
	Аргументы командной строки.
	Выходные данные:
	Число или сообщение об ошибке.
	Решение:
	import argparse
	
	operations = {
	  'add': lambda a, b: a + b,
	  'sub': lambda a, b: a - b,
	  'mul': lambda a, b: a * b,
	  'div': lambda a, b: a / b
	}
	
	parser = argparse.ArgumentParser()
	parser.add_argument('a', type=int)
	parser.add_argument('b', type=int)
	parser.add_argument('-op', choices=['add','sub','mul','div'], default='add')

	parser.add_argument('-round', action='store_true')
	args = parser.parse_args()
	
	try:
	  result = operations[args.op](args.a, args.b)
	  if args.round:
	  print(int(round(result)))
	  else:
	  print(result)
	except (ZeroDivisionError, KeyError):
	  print('Error')

Модуль 4, WEB и API, Использование библиотек	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи
Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполняет не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Мариона Виктория Вадимовна
серийный номер 01ab5191c94fe2024e0127e80 5e8ab
срок действия 05.12.2025 - 05.03.2027

Варианты заданий	1. Поиск всех email-адресов в тексте Напиши программу, которая считывает строку текста и находит в ней все email-адреса. Email-адрес считается корректным, если он содержит символ @, до и после которого есть как минимум один символ (буква, цифра, точка или дефис). Выведи все найденные email-адреса, каждый на новой строке. Если адресов нет, программа не должна ничего выводить. Входные данные: Одна строка текста. Выходные данные: Все найденные email-адреса, каждый на отдельной строке. Пример ввода: Мои контакты: user@example.com, support@site.ru, и test@mail.com Пример вывода: user@example.com support@site.ru test@mail.com Решение: import re text = input() emails = re.findall(r'[\w.-]+@[\w.-]+', text) for email in emails: print(email) 2. Замена дат в тексте на формат ДД.ММ.ГГГГ Напиши программу, которая заменяет все даты в формате ГГГГ-ММ-ДД на формат ДД.ММ.ГГГГ. Используй регулярные выражения для поиска и замены. Дата состоит из 4 цифр года, 2 цифр месяца, 2 цифр дня, разделённых дефисами. Выведи полученную строку. Входные данные: Одна строка текста. Выходные данные: Строка с заменёнными датами. Пример ввода: Сегодня 2025-12-25, а завтра 2025-12-26. Пример вывода: Сегодня 25.12.2025, а завтра 26.12.2025. Решение: import re text = input() result = re.sub(r'(\d{4})-(\d{2})-(\d{2})', r'\3.\2.\1', text) print(result) 3. Функция для подсчёта среднего арифметического уникальных чётных чисел (с assert) Напиши функцию avg_even_unique(lst), которая принимает список целых чисел и возвращает среднее арифметическое тех чётных чисел, которые встречаются в списке ровно один раз. Если таких чисел нет, верни 0. Результат округли до одного знака после запятой. Напиши как минимум 3 assert-теста для проверки функции. Считай список чисел с клавиатуры (через пробел) и выведи результат работы функции.

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Входные данные:
Строка целых чисел через пробел.
Выходные данные:
Одно число (с одним знаком после запятой) или 0.
Пример ввода:
1 2 2 4 4 5 6 6 7
Пример вывода:
0
(чётные числа, встречающиеся ровно один раз: 2 и 6 – два числа? В списке 2 и 6 встречаются по два раза, поэтому уникальных чётных чисел нет, результат 0)
Решение:

```
def avg_even_unique(lst):
    counts = {}
    for x in lst:
        counts[x] = counts.get(x, 0) + 1
    unique_even = [x for x in counts if counts[x] == 1 and x % 2 == 0]
    if not unique_even:
        return 0
    return round(sum(unique_even) / len(unique_even), 1)

# assert-тесты
assert avg_even_unique([1, 2, 3, 4, 5]) == 3.0
assert avg_even_unique([1, 2, 2, 4, 4, 6]) == 6.0
assert avg_even_unique([1, 3, 5]) == 0

lst = list(map(int, input().split()))
print(avg_even_unique(lst))
```

4. Функция с doctest для нахождения пересечения двух списков
Напиши функцию intersection_avg(lst1, lst2), которая принимает два списка чисел и возвращает среднее арифметическое чисел, присутствующих в обоих списках (пересечение). Если пересечение пусто, верни 0. Результат округли до одного знака после запятой. Добавь докстринг с примерами использования (doctest). Считай два списка с клавиатуры (каждый на отдельной строке) и выведи результат.
Входные данные:
Две строки целых чисел через пробел.
Выходные данные:
Одно число (с одним знаком после запятой) или 0.
Пример ввода:
1 2 3 4 5
4 5 6 7 8
Пример вывода:
4.5

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Решение:

```
def intersection_avg(lst1, lst2):
    """
    >>> intersection_avg([1, 2, 3, 4, 5], [4, 5, 6, 7, 8])
    4.5
    >>> intersection_avg([1, 2, 3], [4, 5, 6])
    0
    >>> intersection_avg([1, 2, 2, 3], [2, 2, 4])
    2.0
    """
    inter = set(lst1) & set(lst2)
    if not inter:
        return 0
    return round(sum(inter) / len(inter), 1)

if __name__ == "__main__":
    import doctest
    doctest.testmod()

lst1 = list(map(int, input().split()))
lst2 = list(map(int, input().split()))
print(intersection_avg(lst1, lst2))
```

5. Unittest для проверки функции нахождения наиболее частого элемента

Напиши функцию most_frequent(lst), которая принимает список и возвращает элемент, встречающийся в нём наибольшее число раз. Если таких элементов несколько, верни первый из них в порядке появления в списке. Напиши unittest-тесты для этой функции (в отдельном файле или в том же файле с использованием unittest.main()). Считай список с клавиатуры (через пробел) и выведи результат.

Входные данные:

Строка элементов (чисел или строк) через пробел.

Выходные данные:

Один элемент (строка или число).

Пример ввода:

a b a c d a b c

Пример вывода:

a

Решение:

main.py:

```
def most_frequent(lst):
    counts = {}
    for item in lst:
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

	counts[item] = counts.get(item, 0) + 1
	max_count = max(counts.values())
	for item in lst:
	if counts[item] == max_count:
	return item
	
	lst = input().split()
	print(most_frequent(lst))
	test_main.py:
	import unittest
	from main import most_frequent
	
	class TestMostFrequent(unittest.TestCase):
	def test_simple_list(self):
	self.assertEqual(most_frequent(['a', 'b', 'a', 'c']), 'a')
	
	def test_tie(self):
	self.assertEqual(most_frequent(['a', 'b', 'c', 'b', 'a']), 'a')
	
	def test_numbers(self):
	self.assertEqual(most_frequent([1, 2, 2, 3, 3]), 2)
	
	def test_single_element(self):
	self.assertEqual(most_frequent(['x']), 'x')
	
	if __name__ == '__main__':
	unittest.main()

Модуль 4, WEB и API, WEB и API	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи
Диагностические инструменты	Комплекты из пяти практических заданий по теме Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальная сумма баллов за одну проверочную работу составляет 5 баллов. Задание считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью совпадать с ожидаемым результатом для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Обучающийся подтверждает освоение текущего материала, если выполняет не менее 60% обязательных заданий (то есть правильно решает 3 из 5 задач)

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Варианты заданий	1. Количество вакансий по ключевому слову в регионе (API «Работа России») Напиши программу, которая делает запрос к API «Работа России» для региона с кодом 16 (Республика Татарстан) с параметром text='аналитик'. Из полученного JSON-ответа извлеки количество найденных вакансий (поле results.vacancies – список) и выведи это число. Входные данные: Нет (регион и текст фиксированы). Выходные данные: Одно целое число – количество вакансий. Пример вывода: 24 Решение: import requests url = 'http://opendata.trudvsem.ru/api/v1/vacancies/region/16' params = {'text': 'аналитик'} response = requests.get(url, params=params) data = response.json() vacancies = data['results']['vacancies'] print(len(vacancies)) 2. Информация о персонаже из SWAPI (поиск по имени) Напиши программу, которая запрашивает у пользователя имя персонажа из вселенной Star Wars, отправляет GET-запрос к API SWAPI (https://swapi.dev/api/people/) с параметром search, и выводит количество фильмов, в которых этот персонаж появлялся (поле films – список). Если персонаж не найден, программа выводит «Не найден». Используй обработку случая, когда список результатов пуст. Входные данные: Одна строка – имя персонажа. Выходные данные: Целое число или сообщение «Не найден». Пример ввода: Luke Skywalker Пример вывода: 5 Решение: import requests name = input() url = 'https://swapi.dev/api/people/' params = {'search': name} response = requests.get(url, params=params) data = response.json() if data['results']: films = data['results'][0]['films']

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
    print(len(films))
else:
    print('Не найден')
```

3. Определение города по IP-адресу (API DaData)

Напиши программу, которая запрашивает IP-адрес (например, 46.226.227.20) и использует API DaData «Город по IP-адресу» (POST-запрос к https://suggestions.dadata.ru/suggestions/api/4_1/rs/iplocate/address). Для авторизации используй тестовый токен "test_token". Из JSON-ответа извлеки название города (поле location.data.city). Если город не определён, выведи «Не определён».

Входные данные:

Одна строка – IP-адрес.

Выходные данные:

Название города или сообщение.

Пример ввода:

46.226.227.20

Пример вывода:

г Москва

Решение:

```
import requests
import json

token = "test_token"
ip = input()
url = "https://suggestions.dadata.ru/suggestions/api/4_1/rs/iplocate/address"
headers = {
    "Content-Type": "application/json",
    "Accept": "application/json",
    "Authorization": f"Token {token}"
}
data = {"ip": ip}
response = requests.post(url, headers=headers, data=json.dumps(data))
result = response.json()
if result.get('location'):
    city = result['location']['data']['city']
    print(city)
else:
    print("Не определён")
```

4. Парсинг количества PEP в секции «Open PEPs»

Напиши программу, которая парсит страницу https://peps.python.org/ и подсчитывает количество PEP в секции с идентификатором open-peps. Внутри этой секции все PEP перечислены в таблице. Подсчитай количество строк <tr> в таблице и выведи это число (без учёта заголовка, то есть только строки с данными).

Входные данные:

Нет.

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

	Выходные данные:
	Одно целое число.
	Пример вывода:
	(зависит от актуального состояния страницы)
	Решение:
	import requests
	from bs4 import BeautifulSoup
	
	url = 'https://peps.python.org/'
	response = requests.get(url)
	soup = BeautifulSoup(response.text, 'html.parser')
	section = soup.find('section', id='open-peps')
	table = section.find('table')
	rows = table.find_all('tr')
	# Исключаем первую строку (заголовок)
	print(len(rows) - 1)
	5. Поиск ссылок на стандартные функции Python, начинающихся с 's'
	Напиши программу, которая парсит страницу документации Python https://docs.python.org/3/library/functions.html и выводит названия функций (без аргументов, только имена), которые начинаются с буквы 's' (строчной) в том порядке, в котором они встречаются на странице. Каждое название выводи на отдельной строке.
	Входные данные:
	Нет.
	Выходные данные:
	Список названий функций, каждое на новой строке.
	Пример вывода (фрагмент):
	setattr
	slice
	sorted
	...
	Решение:
	import requests
	from bs4 import BeautifulSoup
	
	url = 'https://docs.python.org/3/library/functions.html'
	response = requests.get(url)
	soup = BeautifulSoup(response.text, 'html.parser')
	for div in soup.find_all('div', class_='line'):
	a = div.find('a')
	if a:

	name = a.text if name.startswith('s'): print(name)
Модуль 4, WEB и API, Проект	
Количество академических часов	
Формы контроля	
Диагностические инструменты	
Показатели и критерии оценивания	
Модуль 4, WEB и API, Кейс-задача	
Количество академических часов	
Формы контроля	
Диагностические инструменты	
Показатели и критерии оценивания	
Промежуточная аттестация (по каждому модулю)	
Основы Python, Модуль 1	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде. Практические задания на написание программного кода по предложенному алгоритму и описанию задачи. Задания на определение результата работы программного кода. Задания открытого типа.

Контур Крипто
владелец Маршеня Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
срок действия 05.12.2025 - 05.03.2027

Диагностические инструменты	Комплекты из 15 практических заданий по теме Не менее 50% заданий составляют практико-ориентированные задания на написание программного кода Не менее 20% заданий составляют задания открытого типа Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальное количество баллов — 15. Задание на написание программного кода считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью соответствовать ожидаемому результату для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Задания на исправление ошибок, определение результата работы программы и задания открытого типа считаются выполненными верно, если полученный ответ полностью соответствует эталонному ответу. Промежуточная аттестация считается пройденной при выполнении не менее 60% заданий (не менее 9 из 15 баллов). Прохождение промежуточной аттестации по модулю является обязательным условием для допуска обучающегося к итоговой аттестации.

Варианты заданий	Задание 1 (написание кода)
	Напиши программу, которая запрашивает радиус круга (вещественное число) и с помощью константы math.pi вычисляет длину окружности и площадь круга. Выведи оба значения на отдельных строках без округления.
	Входные данные:
	Одно вещественное число – радиус.
	Выходные данные:
	Два вещественных числа, каждое на новой строке.
	Пример ввода:
	3
	Пример вывода:
	18.84955592153876
	28.274333882308138
	Возможное решение
	import math
	r = float(input())
	print(2 * math.pi * r)
	print(math.pi * r ** 2)
	
	Задание 2 (исправление ошибок)
	Дана программа, которая должна проверять, является ли введённое число чётным, и выводить соответствующее сообщение:
	x = int(input())
	if x % 2 == 0
	  print('Чётное')
	else
	  print('Нечётное')
	Программа содержит синтаксические ошибки. Исправь их так, чтобы она работала верно.
	Входные данные:
	Одно целое число.
	Выходные данные:
	Строка «Чётное» или «Нечётное».
	Пример ввода:
	4
	Пример вывода:
	Чётное
	Возможное решение:
	x = int(input())
	if x % 2 == 0:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
    print('Чётное')
else:
    print('Нечётное')
```

Задание 3 (написание кода)
Напиши программу, которая запрашивает четырёхзначное целое число и определяет, является ли оно палиндромом (читается одинаково слева направо и справа налево). Если да – выведи «Палиндром», иначе – «Не палиндром».
Входные данные:
Одно целое число (1000–9999).
Выходные данные:
Строка «Палиндром» или «Не палиндром».
Пример ввода:
1221
Пример вывода:
Палиндром
Возможное решение:

```
n = int(input())
a = n // 1000
b = n // 100 % 10
c = n // 10 % 10
d = n % 10
if a == d and b == c:
    print('Палиндром')
else:
    print('Не палиндром')
```

Задание 4 (исправление ошибок)
Дана программа, которая должна выводить все чётные числа от 0 до 10 включительно:

```
for i in range(0, 10, 2):
    print(i)
```

Программа работает, но она не включает 10. Исправь ошибку, чтобы выводились числа 0, 2, 4, 6, 8, 10.
Входные данные:
Ничего не вводится.
Выходные данные:
Числа, каждое на новой строке.
Пример вывода:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

0
2
4
6
8
10
Возможное решение:

```
for i in range(0, 11, 2):
    print(i)
```

Задание 5 (написание кода)
Напиши программу, которая запрашивает у пользователя количество слов n, затем сами слова (каждое на отдельной строке) и выводит аббревиатуру из первых букв каждого слова, переведённых в верхний регистр.
Входные данные:
Первая строка – целое число n ($1 \le n \le 10$).
Затем n строк – слова (состоят только из букв).
Выходные данные:
Одна строка – аббревиатура.
Пример ввода:
3
здоровый
образ
жизни
Пример вывода:
ЗОЖ
Возможное решение:

```
n = int(input())
abbr = ''
for _ in range(n):
    word = input()
    abbr += word[0].upper()
print(abbr)
```

Задание 6 (исправление ошибок)
Дана программа, которая должна с помощью filter и lambda оставить только числа, кратные 3:

```
numbers = list(map(int, input().split()))
result = filter(lambda x: x % 3, numbers)
print(list(result))
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Но она оставляет числа, не кратные 3. Исправь условие в лямбда-функции.
Входные данные:
Строка целых чисел через пробел.
Выходные данные:
Список чисел, кратных 3.
Пример ввода:
1 3 5 6 9 10
Пример вывода:
[3, 6, 9]
Возможное решение:

```
numbers = list(map(int, input().split()))
result = filter(lambda x: x % 3 == 0, numbers)
print(list(result))
```

Задание 7 (написание кода)
Напиши программу, которая запрашивает строку слов, разделённых пробелами, и сортирует их сначала по длине (от коротких к длинным), а при равной длине – по алфавиту. Выведи отсортированные слова через пробел. Используй sorted и lambda.
Входные данные:
Строка слов через пробел.
Выходные данные:
Строка отсортированных слов через пробел.
Пример ввода:
кот собака мышь слон енот лев
Пример вывода:
кот лев слон енот мышь собака
Возможное решение:

```
words = input().split()
sorted_words = sorted(words, key=lambda w: (len(w), w))
print(' '.join(sorted_words))
```

Задание 8 (исправление ошибок)
Дана рекурсивная функция, которая должна вычислять факториал числа:

```
def factorial(n):
    if n == 0:
        return 1
    return factorial(n) * n
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Программа зацикливается из-за ошибки в рекурсивном вызове. Исправь её.
Входные данные:
Одно целое число n (n ≥ 0).
Выходные данные:
Одно целое число – факториал n.
Пример ввода:
5
Пример вывода:
120
Возможное решение:

```
def factorial(n):
    if n == 0:
        return 1
    return factorial(n - 1) * n   # вызов с n-1
```

Задание 9 (написание кода)
Напиши программу, которая запрашивает строку и выводит количество различных символов в ней (регистр учитывается). Используй множество.
Входные данные:
Одна строка.
Выходные данные:
Одно целое число.
Пример ввода:
Hello, world!
Пример вывода:
11   # (буквы, запятая, пробел, восклицательный знак – считаем все символы)
Возможное решение:

```
python
s = input()
print(len(set(s)))
```

Задание 10 (исправление ошибок)

```
text = input().split()
freq = {}
for word in text:
    freq[word] = freq.get(word) + 1
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

print(freq)
Дана программа, которая должна подсчитывать частоту слов, но содержит ошибку. Найди и исправь её.
Входные данные:
Строка слов через пробел.
Выходные данные:
Словарь с частотами слов.
Пример ввода:
яблоко груша яблоко
Пример вывода:
{'яблоко': 2, 'груша': 1}
Возможное решение:

```
text = input().split()
freq = {}
for word in text:
    freq[word] = freq.get(word, 0) + 1
print(freq)
```

Задание 11 (написание кода)
Напиши программу, которая запрашивает строку, содержащую слова и целые числа, разделённые пробелами. Найди среди всех целых чисел то, у которого сумма цифр максимальна. Если таких чисел несколько, выведи первое по порядку. Гарантируется, что хотя бы одно число есть. Используй max с key и filter.
Входные данные:
Строка из слов и чисел через пробел.
Выходные данные:
Одно целое число (исходное, не сумма цифр).
Пример ввода:
abc 123 def 45 ghi 6789
Пример вывода:
6789
Возможное решение:

```
parts = input().split()
numbers = list(filter(lambda x: x.isdigit(), parts))
result = max(numbers, key=lambda n: sum(map(int, n)))
print(result)
```

Задание 12 (краткий ответ)
Вопрос:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

	Какой метод списка добавляет элемент в конец списка? Запиши название метода без скобок.
	Ответ: append
	
	Задание 13 (краткий ответ)
	Вопрос:
	Какое ключевое слово в Python используется для возврата значения из функции?
	Ответ: return
	
	Задание 14 (краткий ответ)
	Вопрос:
	Что возвращает встроенная функция len() при передаче ей строки?
	Ответ: количество символов в строке
	
	Задание 15 (написание кода)
	Напиши программу, которая запрашивает два целых числа (каждое на отдельной строке) и с помощью алгоритма Евклида (через вычитание) находит их наибольший общий делитель (НОД). Выведи НОД.
	Входные данные:
	Два целых числа, каждое на отдельной строке.
	Выходные данные:
	Одно целое число – НОД.
	Пример ввода:
	30
	12
	Пример вывода:
	6
	Возможное решение:
	a = int(input())
	b = int(input())
	while a != 0 and b != 0:
	  if a > b:
	  a = a - b
	  else:
	  b = b - a
	print(a + b)

Шкала оценивания, нижнее значение	0
Шкала оценивания, верхнее значение	15
Шкала оценивания, минимальный проходной балл	9
Объектно-ориентированное программирование (ООП), Модуль 2	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде. Практические задания на написание программного кода по предложенному алгоритму и описанию задачи. Задания на определение результата работы программного кода. Задания открытого типа.
Диагностические инструменты	Комплекты из 15 практических заданий по теме Не менее 50% заданий составляют практико-ориентированные задания на написание программного кода Не менее 20% заданий составляют задания открытого типа Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальное количество баллов — 15. Задание на написание программного кода считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью соответствовать ожидаемому результату для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Задания на исправление ошибок, определение результата работы программы и задания открытого типа считаются выполненными верно, если полученный ответ полностью соответствует эталонному ответу. Промежуточная аттестация считается пройденной при выполнении не менее 60% заданий (не менее 9 из 15 баллов). Прохождение промежуточной аттестации по модулю является обязательным условием для допуска обучающегося к итоговой аттестации.

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Варианты заданий	Задание 1 (написание кода)
	Создай класс Book с атрибутами: title (название), author (автор), pages (количество страниц), is_read (прочитана ли книга – bool, по умолчанию False). Реализуй:
	конструктор __init__(title, author, pages);
	метод read(), который отмечает книгу как прочитанную (is_read = True);
	метод info(), который выводит строку в формате:
	«{автор} --- {название} ({страницы} стр.) --- {“Прочитана” if is_read else “Не прочитана”}».
	В основной программе создай два экземпляра класса Book с данными, введёнными с клавиатуры (для каждой книги: название, автор, количество страниц). Для второй книги вызови метод read(). Выведи информацию о каждой книге.
	Входные данные:
	Шесть строк: название первой книги, автор первой книги, количество страниц первой книги, название второй книги, автор второй книги, количество страниц второй книги.
	Выходные данные:
	Две строки – результат вызова метода info().
	Пример ввода:
	Война и мир
	Толстой
	1425
	1984
	Оруэлл
	328
	Пример вывода:
	Толстой --- Война и мир (1425 стр.) --- Не прочитана
	Оруэлл --- 1984 (328 стр.) --- Прочитана
	Возможное решение:
	class Book:
	def __init__(self, title, author, pages):
	self.title = title
	self.author = author
	self.pages = pages
	self.is_read = False
	
	def read(self):
	self.is_read = True
	
	def info(self):
	status = "Прочитана" if self.is_read else "Не прочитана"
	print(f'{self.author} --- {self.title} ({self.pages} стр.) --- {status}')
	
	title1 = input()

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
author1 = input()
pages1 = int(input())
book1 = Book(title1, author1, pages1)

title2 = input()
author2 = input()
pages2 = int(input())
book2 = Book(title2, author2, pages2)
book2.read()

book1.info()
book2.info()
```

Задание 2 (написание кода)
Создай класс Product с атрибутами: name (название) и price (цена – целое число). Реализуй:
конструктор __init__(name, price);
статический метод is_affordable(price, budget), который возвращает True, если price <= budget, и False в противном случае;
метод класса from_string(data), который принимает строку в формате «название: цена» (например, «Телефон: 50000»), создаёт и возвращает объект Product;
метод __str__, возвращающий строку «Товар: {name}, цена: {price} руб.».
В основной программе прочитай одну строку в формате «название: цена», создай объект с помощью from_string() и выведи его строковое представление. Затем прочитай бюджет (целое число) и выведи, доступен ли товар по цене.
Входные данные:
Первая строка – «название: цена».
Вторая строка – бюджет (целое число).
Выходные данные:
Строковое представление товара и True/False – доступность по цене.
Пример ввода:
Телефон: 50000
45000
Пример вывода:
Товар: Телефон, цена: 50000 руб.
False
Возможное решение:

```
class Product:
    def __init__(self, name, price):
        self.name = name
        self.price = price
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

```
    @staticmethod
    def is_affordable(price, budget):
        return price <= budget

    @classmethod
    def from_string(cls, data):
        name, price = data.split(': ')
        return cls(name, int(price))

    def __str__(self):
        return f'Товар: {self.name}, цена: {self.price} руб.'

data = input()
budget = int(input())
product = Product.from_string(data)
print(product)
print(Product.is_affordable(product.price, budget))
```

Задание 3 (написание кода)
Создай класс Person (человек) с атрибутами name (строка) и age (целое число). Реализуй методы:
__init__(name, age);
get_name(), возвращающий строку «Человек: {name}»;
get_age(), возвращающий строку «Возраст: {age}».
Создай класс Student (ученик), наследующий от Person. Переопредели:
__init__(name, age, course), где course – целое число (класс);
get_name(), возвращающий строку «Ученик: {name} ({course} класс)».
В основной программе прочитай три строки: имя, возраст и класс. Создай объект Student и выведи результаты методов get_name() и get_age() на разных строках.
Входные данные:
Три строки: имя, возраст (целое число), класс (целое число).
Выходные данные:
Две строки.
Пример ввода:
Иванов Иван
15
7

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Пример вывода:
Ученик: Иванов Иван (7 класс)
Возраст: 15
Возможное решение:

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def get_name(self):
        return f'Человек: {self.name}'

    def get_age(self):
        return f'Возраст: {self.age}'

class Student(Person):
    def __init__(self, name, age, course):
        super().__init__(name, age)
        self.course = course

    def get_name(self):
        return f'Ученик: {self.name} ({self.course} класс)'

name = input()
age = int(input())
course = int(input())
student = Student(name, age, course)
print(student.get_name())
print(student.get_age())
```

Задание 4 (написание кода)
Создай четыре базовых класса игровых персонажей:
Warrior с методом attack(), возвращающим «Атака мечом»;
Mage с методом cast_spell(), возвращающим «Магическое заклинание»;
Archer с методом shoot(), возвращающим «Выстрел из лука»;
Healer с методом heal(), возвращающим «Лечение союзника».

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Создай уникальных персонажей с использованием множественного наследования:
Paladin наследуется от Warrior и Healer;
BattleMage наследуется от Warrior и Mage;
Ranger наследуется от Archer и Healer.
В основной программе прочитай название уникального персонажа (Paladin, BattleMage или Ranger). Создай объект соответствующего класса и выведи все его доступные действия, каждое на отдельной строке.
Входные данные:
Одна строка – название уникального персонажа.
Выходные данные:
Действия выбранного персонажа, каждое на отдельной строке.
Пример ввода:
Paladin
Пример вывода:
Атака мечом
Лечение союзника
Возможное решение:

```
class Warrior:
    def attack(self):
        return 'Атака мечом'

class Mage:
    def cast_spell(self):
        return 'Магическое заклинание'

class Archer:
    def shoot(self):
        return 'Выстрел из лука'

class Healer:
    def heal(self):
        return 'Лечение союзника'

class Paladin(Warrior, Healer):
    pass

class BattleMage(Warrior, Mage):
    pass
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
class Ranger(Archer, Healer):
    pass

hero_type = input()
if hero_type == 'Paladin':
    hero = Paladin()
    print(hero.attack())
    print(hero.heal())
elif hero_type == 'BattleMage':
    hero = BattleMage()
    print(hero.attack())
    print(hero.cast_spell())
elif hero_type == 'Ranger':
    hero = Ranger()
    print(hero.shoot())
    print(hero.heal())
```

Задание 5 (написание кода)
Создай класс Fraction для работы с обыкновенными дробями. Конструктор принимает числитель и знаменатель (целые числа). Реализуй:
метод __str__, возвращающий строку «Дробь num/den» (несократимую дробь);
метод __add__ для сложения двух дробей, возвращающий новый объект Fraction (сокращённый);
статический метод gcd(a, b) для нахождения наибольшего общего делителя (используй алгоритм Евклида) – можно использовать math.gcd.
В основной программе прочитай две строки, каждая содержит два целых числа – числитель и знаменатель первой и второй дроби. Выведи сумму этих дробей.
Входные данные:
Две строки, в каждой – два целых числа через пробел.
Выходные данные:
Одна строка – сумма дробей в формате «Дробь num/den».
Пример ввода:
1 2
2 3
Пример вывода:
Дробь 7/6
Возможное решение:

```
import math

class Fraction:
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
    def __init__(self, num, den):
        self.num, self.den = self.reduce(num, den)

    @staticmethod
    def reduce(num, den):
        g = math.gcd(abs(num), abs(den))
        return num // g, den // g

    def __add__(self, other):
        new_num = self.num * other.den + other.num * self.den
        new_den = self.den * other.den
        return Fraction(new_num, new_den)

    def __str__(self):
        return f'Дробь {self.num}/{self.den}'

n1, d1 = map(int, input().split())
f1 = Fraction(n1, d1)
n2, d2 = map(int, input().split())
f2 = Fraction(n2, d2)
print(f1 + f2)
```

Задание 6 (написание кода)
Создай класс Temperature с приватным атрибутом __celsius. Реализуй:
конструктор __init__(celsius), который устанавливает начальную температуру, но если значение меньше -273.15, выводит «Недопустимая температура, установлено 0» и устанавливает 0;
геттер celsius через @property, возвращающий значение;
сеттер celsius, который позволяет изменить значение, только если оно >= -273.15; иначе выводит «Температура не может быть ниже абсолютного нуля» и не меняет значение;
метод __str__, возвращающий строку «Температура: {значение} °C».
В основной программе прочитай два числа (начальная температура и новая температура). Создай объект с начальной температурой, затем попробуй установить новую температуру через сеттер, выведи строковое представление объекта.
Входные данные:
Два целых числа (начальная и новая температура) через пробел.
Выходные данные:
Сообщения об ошибках (если были) и строковое представление.
Пример ввода:
-300 25
Пример вывода:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Недопустимая температура, установлено 0
Температура: 25 °C
Возможное решение:

```
class Temperature:
    def __init__(self, celsius):
        self.__celsius = 0
        self.celsius = celsius

    @property
    def celsius(self):
        return self.__celsius

    @celsius.setter
    def celsius(self, value):
        if value < -273.15:
            print('Недопустимая температура, установлено 0')
            self.__celsius = 0
        else:
            self.__celsius = value

    def __str__(self):
        return f'Температура: {self.__celsius} °C'

start, new = map(int, input().split())
t = Temperature(start)
t.celsius = new
print(t)
```

Задание 7 (написание кода)
Создай класс Password. Конструктор принимает строку с паролем и выполняет проверку:
длина пароля не менее 8 символов (иначе вызывается исключение MinLengthError);
пароль должен содержать хотя бы один символ из набора !@#$%^&*()<>?,. (иначе вызывается исключение NoSpecialCharError).
Если пароль корректен, он сохраняется в атрибут __password. Если нет – программа должна вывести соответствующее сообщение («Короткий пароль» или «Нет специального символа»).
Создай собственные классы исключений, унаследованные от Exception.
В основной программе прочитай одну строку – пароль. Создай объект класса Password с этим паролем. Если пароль корректен, выведи «Пароль: {пароль}».
Входные данные:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Одна строка – пароль.
Выходные данные:
Сообщение об ошибке или «Пароль: ...».
Пример ввода:
12345678
Пример вывода:
Нет специального символа
Пример ввода 2:
12345678@
Пример вывода 2:
Пароль: 12345678@
Возможное решение:

```
class MinLengthError(Exception):
    pass

class NoSpecialCharError(Exception):
    pass

class Password:
    SPECIAL = '!@#$%^&*()<>?,.'

    def __init__(self, password):
        if len(password) < 8:
            raise MinLengthError('Короткий пароль')
        if not any(ch in self.SPECIAL for ch in password):
            raise NoSpecialCharError('Нет специального символа')
        self.__password = password

    def __str__(self):
        return f'Пароль: {self.__password}'

s = input()
try:
    p = Password(s)
    print(p)
except MinLengthError as e:
    print(e)
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
except NoSpecialCharError as e:
    print(e)
```

Задание 8 (написание кода)
Создай класс Cylinder (цилиндр) с атрибутами radius и height (вещественные числа). Реализуй метод volume(), возвращающий объём цилиндра по формуле π * r² * h (используй math.pi).
Создай класс Box (прямоугольный параллелепипед) с атрибутами length, width, height. Реализуй метод volume(), возвращающий объём length * width * height.
В основной программе прочитай две строки:
первая – тип фигуры (Cylinder или Box);
вторая – три числа через пробел (для цилиндра: радиус и высота; для параллелепипеда: длина, ширина, высота).
Создай объект соответствующего класса и выведи его объём, округлённый до двух знаков после запятой.
Входные данные:
Две строки: тип фигуры и три числа через пробел.
Выходные данные:
Одно вещественное число с двумя знаками после запятой.
Пример ввода:
Cylinder
3 5
Пример вывода:
141.37
Возможное решение:

```
import math

class Cylinder:
    def __init__(self, radius, height):
        self.radius = radius
        self.height = height

    def volume(self):
        return math.pi * self.radius ** 2 * self.height

class Box:
    def __init__(self, length, width, height):
        self.length = length
        self.width = width
        self.height = height
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

```
    def volume(self):
        return self.length * self.width * self.height

t = input()
vals = list(map(float, input().split()))
if t == 'Cylinder':
    obj = Cylinder(vals[0], vals[1])
else:
    obj = Box(vals[0], vals[1], vals[2])
print(round(obj.volume(), 2))
```

Задание 9 (исправление ошибок)
Дана программа, которая должна создавать класс Student с атрибутами name и course, и методом info(), выводящим информацию о студенте. В коде есть ошибка. Найди и исправь её.

```
class Student:
    def __init__(name, course):
        self.name = name
        self.course = course

    def info():
        print(f'Студент: {self.name}, курс {self.course}')

s = Student('Анна', 2)
s.info()
```

Входные данные:
Ничего не вводится.
Выходные данные:
Строка.
Пример вывода:
Студент: Анна, курс 2
Возможное решение:

```
class Student:
    def __init__(self, name, course):   # добавлен self
        self.name = name
        self.course = course

    def info(self):                     # добавлен self
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

```
        print(f'Студент: {self.name}, курс {self.course}')

s = Student('Анна', 2)
s.info()
```

Задание 10 (исправление ошибок)
Дана программа с наследованием. Класс Vehicle – базовый, класс Car – наследник. Программа должна выводить «Автомобиль: двигатель запущен», но содержит ошибку. Исправь её.

```
class Vehicle:
    def __init__(self, name):
        self.name = name

    def start_engine(self):
        return 'Двигатель запущен'

class Car(Vehicle):
    def __init__(self, name):
        self.name = name

    def start_engine(self):
        return f'Автомобиль {self.name}: двигатель запущен'

car = Car('Tesla')
print(car.start_engine())
```

Программа работает, но нарушает принцип наследования: в Car дублируется инициализация name. Исправь, используя super().
Входные данные:
Ничего не вводится.
Выходные данные:
Строка.
Пример вывода:
Автомобиль Tesla: двигатель запущен
Возможное решение:

```
class Vehicle:
    def __init__(self, name):
        self.name = name

    def start_engine(self):
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

```
        return 'Двигатель запущен'

class Car(Vehicle):
    def __init__(self, name):
        super().__init__(name)   # вызов конструктора родителя

    def start_engine(self):
        return f'Автомобиль {self.name}: двигатель запущен'

car = Car('Tesla')
print(car.start_engine())
```

Задание 11 (исправление ошибок)
Дана программа, которая использует инкапсуляцию для хранения возраста. Код содержит ошибку при попытке изменения приватного атрибута. Исправь программу так, чтобы она работала корректно.

```
class Person:
    def __init__(self, age):
        self.__age = age

    def get_age(self):
        return self.__age

    def set_age(self, age):
        if age > 0:
            self.__age = age
        else:
            print('Некорректный возраст')

p = Person(25)
p.__age = 30
print(p.get_age())
```

Входные данные:
Ничего не вводится.
Выходные данные:
Число 30
Пример вывода:
30

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Возможное решение:

```
class Person:
    def __init__(self, age):
        self.__age = age

    def get_age(self):
        return self.__age

    def set_age(self, age):
        if age > 0:
            self.__age = age
        else:
            print('Некорректный возраст')

p = Person(25)
p.set_age(30)   # исправлено: использовать сеттер
print(p.get_age())
```

Задание 12 (исправление ошибок)

Дана программа с перегрузкой оператора + для класса Vector. Она должна складывать два вектора и возвращать новый вектор. В коде есть ошибка. Исправь её.

```
class Vector:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def __add__(self, other):
        return self.x + other.x, self.y + other.y   # возвращает кортеж, а не Vector

    def __str__(self):
        return f'Vector({self.x}, {self.y})'

v1 = Vector(1, 2)
v2 = Vector(3, 4)
v3 = v1 + v2
print(v3)
```

Входные данные:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

	Ничего не вводится.
	Выходные данные:
	Vector(4, 6)
	Возможное решение:
	class Vector:
	  def __init__(self, x, y):
	  self.x = x
	  self.y = y
	
	  def __add__(self, other):
	  return Vector(self.x + other.x, self.y + other.y) # исправлено
	
	  def __str__(self):
	  return f'Vector({self.x}, {self.y})'
	
	v1 = Vector(1, 2)
	v2 = Vector(3, 4)
	v3 = v1 + v2
	print(v3)
	
	Задание 13 (краткий ответ)
	Вопрос:
	Какой специальный метод вызывается при создании нового объекта класса (конструктор)? Запиши название метода без скобок.
	Ответ: __init__
	
	Задание 14 (краткий ответ)
	Вопрос:
	Какой декоратор используется для создания геттера (свойства) в Python? Запиши название декоратора.
	Ответ: @property
	
	Задание 15 (краткий ответ)
	Вопрос:
	Что необходимо сделать, чтобы создать собственный класс исключения в Python? Запиши одним предложением.
	Ответ: унаследовать от Exception

Шкала оценивания, нижнее значение	0
Шкала оценивания, верхнее значение	15
Шкала оценивания, минимальный проходной балл	9
Алгоритмы и структуры данных, Модуль 3	
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде. Практические задания на написание программного кода по предложенному алгоритму и описанию задачи. Задания на определение результата работы программного кода. Задания открытого типа.
Диагностические инструменты	Комплекты из 15 практических заданий по теме Не менее 50% заданий составляют практико-ориентированные задания на написание программного кода Не менее 20% заданий составляют задания открытого типа Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальное количество баллов — 15. Задание на написание программного кода считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью соответствовать ожидаемому результату для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Задания на исправление ошибок, определение результата работы программы и задания открытого типа считаются выполненными верно, если полученный ответ полностью соответствует эталонному ответу. Промежуточная аттестация считается пройденной при выполнении не менее 60% заданий (не менее 9 из 15 баллов). Прохождение промежуточной аттестации по модулю является обязательным условием для допуска обучающегося к итоговой аттестации.

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Варианты заданий	Задание 1 (написание кода)
	Создай класс Node и класс LinkedList, который реализует односвязный список. Реализуй метод append(value) – добавляет элемент в конец списка, и метод print_list() – выводит все элементы списка через пробел в порядке их следования. В основной программе прочитай n целых чисел и добавь их в список, затем выведи список с помощью print_list().
	Входные данные:
	Первая строка – n (количество элементов). Вторая строка – n целых чисел через пробел.
	Выходные данные:
	Элементы списка через пробел.
	Пример ввода:
	5
	10 20 30 40 50
	Пример вывода:
	10 20 30 40 50
	Возможное решение:
	class Node:
	  def __init__(self, value):
	  self.value = value
	  self.next = None
	
	class LinkedList:
	  def __init__(self):
	  self.head = None
	  def append(self, value):
	  new_node = Node(value)
	  if self.head is None:
	  self.head = new_node
	  return
	  current = self.head
	  while current.next:
	  current = current.next
	  current.next = new_node
	  def print_list(self):
	  current = self.head
	  while current:
	  print(current.value, end=' ')
	  current = current.next
	
	n = int(input())
	arr = list(map(int, input().split()))

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
lst = LinkedList()
for x in arr:
    lst.append(x)
lst.print_list()
```

Задание 2 (написание кода)
Реализуй класс Stack (стек) на основе списка с методами:
push(item) – добавляет элемент на вершину стека;
pop() – удаляет и возвращает верхний элемент (если стек пуст, возвращает None и выводит «error»);
size() – возвращает количество элементов.
В основной программе прочитай количество команд n, затем n строк команд: push X, pop, size. Для pop выведи результат (каждый на новой строке).
Входные данные:
Первая строка – n, затем n строк команд.
Выходные данные:
Результаты pop.
Пример ввода:
4
push 5
push 10
pop
pop
Пример вывода:
10
5
Возможное решение:

```
class Stack:
    def __init__(self):
        self.items = []
    def push(self, item):
        self.items.append(item)
    def pop(self):
        if not self.items:
            print('error')
            return None
        return self.items.pop()
    def size(self):
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

```
        return len(self.items)

n = int(input())
s = Stack()
for _ in range(n):
    cmd = input().split()
    if cmd[0] == 'push':
        s.push(int(cmd[1]))
    elif cmd[0] == 'pop':
        result = s.pop()
        if result is not None:
            print(result)
    elif cmd[0] == 'size':
        print(s.size())
```

Задание 3 (написание кода)
Реализуй класс Queue (очередь) на основе связного списка с методами:
push(item) – добавляет элемент в конец очереди (O(1));
pop() – удаляет и возвращает первый элемент (если очередь пуста, выводит «error» и возвращает None);
size() – возвращает количество элементов.
В основной программе прочитай команды и выведи результаты pop.
Входные данные:
Первая строка – n, затем n строк команд.
Выходные данные:
Результаты pop.
Пример ввода:
5
push 1
push 2
push 3
pop
pop
Пример вывода:
1
2
Возможное решение:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

```
class Node:
    def __init__(self, value):
        self.value = value
        self.next = None

class Queue:
    def __init__(self):
        self.head = None
        self.tail = None
        self.length = 0
    def push(self, item):
        new_node = Node(item)
        if self.tail is None:
            self.head = self.tail = new_node
        else:
            self.tail.next = new_node
            self.tail = new_node
        self.length += 1
    def pop(self):
        if self.head is None:
            print('error')
            return None
        val = self.head.value
        self.head = self.head.next
        if self.head is None:
            self.tail = None
        self.length -= 1
        return val
    def size(self):
        return self.length

n = int(input())
q = Queue()
for _ in range(n):
    cmd = input().split()
    if cmd[0] == 'push':
        q.push(int(cmd[1]))
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
    elif cmd[0] == 'pop':
        result = q.pop()
        if result is not None:
            print(result)
    elif cmd[0] == 'size':
        print(q.size())
```

Задание 4 (написание кода)
Напиши программу, которая проверяет, является ли введённая строка правильной скобочной последовательностью. Поддерживаются три типа скобок: (), [], {}. Используй стек для проверки. Выведи True, если последовательность правильная, иначе False.
Входные данные:
Одна строка, содержащая только символы ()[]{}.
Выходные данные:
True или False.
Пример ввода:
({[]})
Пример вывода:
True
Возможное решение:

```
s = input()
stack = []
pairs = {')': '(', ']': '[', '}': '{'}
for ch in s:
    if ch in '([{':
        stack.append(ch)
    else:
        if not stack or stack[-1] != pairs[ch]:
            print(False)
            break
        stack.pop()
else:
    print(len(stack) == 0)
```

Задание 5 (написание кода)
Напиши программу, которая сортирует список целых чисел по возрастанию с помощью сортировки вставками. После каждой вставки (т.е. после того, как очередной элемент вставлен на своё место) выводи текущее состояние списка. Выведи n-1 строку (где n – размер списка). Если список уже отсортирован, всё равно выводи состояния для каждого шага.
Входные данные:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Первая строка – n, вторая – n целых чисел через пробел.
Выходные данные:
n-1 строк с элементами списка через пробел после каждой вставки.
Пример ввода:
5
4 2 7 1 3
Пример вывода:
2 4 7 1 3
2 4 7 1 3
1 2 4 7 3
1 2 3 4 7
Возможное решение:

```
n = int(input())
arr = list(map(int, input().split()))
for i in range(1, n):
    key = arr[i]
    j = i - 1
    while j >= 0 and arr[j] > key:
        arr[j + 1] = arr[j]
        j -= 1
    arr[j + 1] = key
    print(' '.join(map(str, arr)))
```

Задание 6 (написание кода)
Напиши программу, которая находит индекс первого вхождения элемента x в отсортированном по неубыванию списке. Используй левый двоичный поиск. Если элемент не найден, выведи -1. Индексация с 0.
Входные данные:
Первая строка – n (размер списка). Вторая строка – n целых чисел (отсортированы по неубыванию). Третья строка – x.
Выходные данные:
Индекс первого вхождения x или -1.
Пример ввода:
8
1 3 3 3 7 9 9 12
3
Пример вывода:
1
Возможное решение:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

```
n = int(input())
arr = list(map(int, input().split()))
x = int(input())
left, right = 0, n
found = -1
while left < right:
    mid = (left + right) // 2
    if arr[mid] < x:
        left = mid + 1
    elif arr[mid] > x:
        right = mid
    else:
        if mid == 0 or arr[mid - 1] != x:
            found = mid
            break
        else:
            right = mid
print(found)
```

Задание 7 (написание кода)
Напиши программу, которая определяет, есть ли в отсортированном по неубыванию массиве два различных элемента, сумма которых равна заданному числу k. Используй алгоритм с двумя указателями (O(n)). Выведи True или False.
Входные данные:
Первая строка – n (размер массива). Вторая строка – n целых чисел (отсортированы по неубыванию). Третья строка – k.
Выходные данные:
True или False.
Пример ввода:
6
1 2 4 7 11 15
15
Пример вывода:
True
Возможное решение:

```
n = int(input())
arr = list(map(int, input().split()))
k = int(input())
left, right = 0, n - 1
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

```
found = False
while left < right:
    s = arr[left] + arr[right]
    if s == k:
        found = True
        break
    elif s < k:
        left += 1
    else:
        right -= 1
print(found)
```

Задание 8 (написание кода)
Создай класс Deque (дек) с методами:
push_back(item) – добавляет элемент в конец;
push_front(item) – добавляет элемент в начало;
pop_back() – удаляет и возвращает последний элемент (если дек пуст, выводит «error» и возвращает None);
pop_front() – удаляет и возвращает первый элемент (если дек пуст, выводит «error» и возвращает None);
size() – возвращает количество элементов.
В основной программе прочитай команды и выведи результаты pop_back и pop_front.
Входные данные:
Первая строка – n, затем n строк команд.
Выходные данные:
Результаты pop_back и pop_front.
Пример ввода:
6
push_front 10
push_back 20
push_front 5
pop_front
pop_back
pop_front
Пример вывода:
5
20
10

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Возможное решение:

```
class Deque:
    def __init__(self):
        self.items = []
    def push_back(self, item):
        self.items.append(item)
    def push_front(self, item):
        self.items.insert(0, item)
    def pop_back(self):
        if not self.items:
            print('error')
            return None
        return self.items.pop()
    def pop_front(self):
        if not self.items:
            print('error')
            return None
        return self.items.pop(0)
    def size(self):
        return len(self.items)

n = int(input())
d = Deque()
for _ in range(n):
    cmd = input().split()
    if cmd[0] == 'push_back':
        d.push_back(int(cmd[1]))
    elif cmd[0] == 'push_front':
        d.push_front(int(cmd[1]))
    elif cmd[0] == 'pop_back':
        result = d.pop_back()
        if result is not None:
            print(result)
    elif cmd[0] == 'pop_front':
        result = d.pop_front()
        if result is not None:
            print(result)
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
    elif cmd[0] == 'size':
        print(d.size())
```

Задание 9 (исправление ошибок)
Программа должна проверять, является ли строка правильной скобочной последовательностью (только круглые скобки). Код содержит ошибку – он не проверяет случай, когда стек пуст, а встретилась закрывающая скобка. Исправь.

```
s = input()
stack = []
for ch in s:
    if ch == '(':
        stack.append(ch)
    elif ch == ')':
        stack.pop()
if stack:
    print(False)
else:
    print(True)
```

Входные данные:
Строка из скобок.
Выходные данные:
True или False.
Пример ввода:
())(
Пример вывода:
False
Возможное решение:

```
s = input()
stack = []
for ch in s:
    if ch == '(':
        stack.append(ch)
    elif ch == ')':
        if not stack:   # добавлена проверка на пустоту
            print(False)
            break
        stack.pop()
else:
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

```
    print(len(stack) == 0)


Задание 10 (исправление ошибок)
Программа должна удалять элемент из связного списка по номеру (нумерация с 0). Код содержит ошибку – при удалении первого элемента не обновляется голова списка. Исправь.
class Node:
    def __init__(self, value, next):
        self.value = value
        self.next = next

def delete_node(head, index):
    node = head
    i = 0
    while i < index - 1:
        node = node.next
        i += 1
    node.next = node.next.next
    return head

# Список: a -> b -> c -> None
n3 = Node('c', None)
n2 = Node('b', n3)
n1 = Node('a', n2)
head = delete_node(n1, 0)  # должно удалить 'a'
Ошибка: при index == 0 цикл не выполняется, и код пытается удалить следующий элемент. Исправь функцию delete_node.
Возможное решение:
def delete_node(head, index):
    if index == 0:                # добавлена обработка первого элемента
        return head.next
    node = head
    i = 0
    while i < index - 1:
        node = node.next
        i += 1
    node.next = node.next.next
    return head
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Задание 11 (исправление ошибок)
Программа должна выполнять сортировку выбором по возрастанию. Код содержит ошибку – он ищет максимальный элемент, но ставит его в начало, что нарушает алгоритм. Исправь, чтобы сортировка работала по возрастанию (минимальный элемент ставится в начало, либо максимальный – в конец). Выбери один из вариантов.

```
n = int(input())
arr = list(map(int, input().split()))
for i in range(n - 1):
    max_idx = 0
    for j in range(1, n - i):
        if arr[j] > arr[max_idx]:
            max_idx = j
    arr[i], arr[max_idx] = arr[max_idx], arr[i]
print(*arr)
```

Пример ввода:

```
5
7 2 9 1 5
```

Ожидаемый вывод (сортировка по возрастанию, после каждого прохода):

```
2 7 9 1 5
1 7 9 2 5
1 2 9 7 5
1 2 5 7 9
```

Возможное решение: (исправлено: ищем максимум и ставим в конец, либо ищем минимум и ставим в начало. Ниже – вариант с минимумом в начало.)

```
n = int(input())
arr = list(map(int, input().split()))
for i in range(n - 1):
    min_idx = i
    for j in range(i + 1, n):
        if arr[j] < arr[min_idx]:
            min_idx = j
    arr[i], arr[min_idx] = arr[min_idx], arr[i]
    print(*arr)
```

Задание 12 (исправление ошибок)
Программа должна вычислять выражение в обратной польской записи. Код содержит ошибку – для операции деления используется целочисленное деление, но из-за неверного порядка операндов результат получается неверным. Исправь.

```
class Stack:
    def __init__(self):
        self.items = []
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

	def push(self, value):
	  self.items.append(value)
	def pop(self):
	  return self.items.pop()
	
	tokens = input().split()
	stack = Stack()
	for t in tokens:
	  if t == '+':
	  a = stack.pop()
	  b = stack.pop()
	  stack.push(a + b)
	  elif t == '-':
	  a = stack.pop()
	  b = stack.pop()
	  stack.push(a - b)
	  elif t == '*':
	  a = stack.pop()
	  b = stack.pop()
	  stack.push(a * b)
	  elif t == '/':
	  a = stack.pop()
	  b = stack.pop()
	  stack.push(a // b)
	  else:
	  stack.push(int(t))
	print(stack.pop())
	Ошибка: при операциях вычитания и деления порядок операндов должен быть b - a и b // a, а не a - b и a // b. Исправь.
	Возможное решение:
	elif t == '-':
	  b = stack.pop()
	  a = stack.pop()
	  stack.push(a - b)
	elif t == '/':
	  b = stack.pop()
	  a = stack.pop()
	  stack.push(a // b)

	Задание 13 (краткий ответ) Вопрос: Какая структура данных работает по принципу LIFO (последним пришёл – первым ушёл)? Запиши одним словом. Ответ: стек Задание 14 (краткий ответ) Вопрос: Какая временная сложность у алгоритма двоичного поиска в отсортированном массиве? Запиши в виде O(…). Ответ: O(logn) Задание 15 (краткий ответ) Вопрос: Какая структура данных позволяет добавлять и удалять элементы с обоих концов? Запиши одним словом. Ответ: дек
Шкала оценивания, нижнее значение	0
Шкала оценивания, верхнее значение	15
Шкала оценивания, минимальный проходной балл	9
	WEB и API, Модуль 4
Количество академических часов	1
Формы контроля	Практические задания на исправление ошибок в готовом программном коде. Практические задания на написание программного кода по предложенному алгоритму и описанию задачи. Задания на определение результата работы программного кода. Задания открытого типа.

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Марцева Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Диагностические инструменты	Комплекты из 15 практических заданий по теме Не менее 50% заданий составляют практико-ориентированные задания на написание программного кода Не менее 20% заданий составляют задания открытого типа Встроенная автоматизированная система проверки кода
Показатели и критерии оценивания	Каждое задание оценивается в 1 балл. Максимальное количество баллов — 15. Задание на написание программного кода считается выполненным верно, если программа успешно проходит все тесты автоматической проверки. Вывод программы должен полностью соответствовать ожидаемому результату для каждого теста (с учётом регистра символов и пробелов внутри строк). Лишние пробелы в конце строки не учитываются. Задания на исправление ошибок, определение результата работы программы и задания открытого типа считаются выполненными верно, если полученный ответ полностью соответствует эталонному ответу. Промежуточная аттестация считается пройденной при выполнении не менее 60% заданий (не менее 9 из 15 баллов). Прохождение промежуточной аттестации по модулю является обязательным условием для допуска обучающегося к итоговой аттестации.

Варианты заданий	Задание 1 (написание кода)
	В файле numbers.txt хранятся целые числа, разделённые пробелами (в нескольких строках). Напиши программу, которая вычисляет сумму всех чисел из файла и выводит её. Используй менеджер контекста with. Если файл пуст, программа должна вывести 0.
	Входные данные:
	Файл numbers.txt с числами.
	Выходные данные:
	Одно целое число – сумма.
	Пример ввода (файл numbers.txt):
	10 20 30
	40 50
	Пример вывода:
	150
	Возможное решение:
	total = 0
	with open('numbers.txt', encoding='utf-8') as f:
	  for line in f:
	  for num in line.split():
	  total += int(num)
	print(total)
	
	Задание 2 (написание кода)
	Скачай файл weather.json (структура: список прогнозов с полем main.temp_min и main.temp_max). Напиши программу, которая находит среднюю температуру (temp_min + temp_max / 2) за весь период и выводит её, округлённую до одного знака после запятой. Если файл не содержит данных, выведи 0.
	Входные данные:
	Файл weather.json.
	Выходные данные:
	Одно вещественное число – средняя температура.
	Пример структуры:
	{
	  "list": [
	  {"main": {"temp_min": 5.2, "temp_max": 7.8}},
	  {"main": {"temp_min": -1.3, "temp_max": 2.5}}
	  ]
	}
	Пример вывода:
	3.6
	((5.2+7.8)/2 = 6.5, (-1.3+2.5)/2 = 0.6, (6.5+0.6)/2 = 3.55 → 3.6)
	Возможное решение:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

```
import json
with open('weather.json', encoding='utf-8') as f:
    data = json.load(f)
if not data['list']:
    print(0)
else:
    avg_sum = 0
    for entry in data['list']:
        avg_sum += (entry['main']['temp_min'] + entry['main']['temp_max']) / 2
    avg = avg_sum / len(data['list'])
    print(round(avg, 1))
```

Задание 3 (написание кода)
В файле students.csv в формате CSV есть столбцы Name и Score (оценка – целое число). Напиши программу, которая находит ученика с самым высоким баллом. Если таких учеников несколько, выведи первого по порядку в файле. Выведи строку в формате:
«Лучший ученик: {имя}, балл: {балл}».
Входные данные:
Файл students.csv.
Выходные данные:
Одна строка.
Пример ввода (файл):
Name,Score
Анна,95
Иван,88
Мария,95
Пример вывода:
Лучший ученик: Анна, балл: 95
Возможное решение:

```
import csv
max_score = -1
max_name = ''
with open('students.csv', newline='', encoding='utf-8') as f:
    reader = csv.DictReader(f)
    for row in reader:
        score = int(row['Score'])
        if score > max_score:
            max_score = score
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

```
            max_name = row['Name']
print(f'Лучший ученик: {max_name}, балл: {max_score}')
```

Задание 4 (написание кода)
С клавиатуры (через sys.stdin) вводится произвольное количество строк, каждая строка содержит слова, разделённые пробелами. Напиши программу, которая группирует слова по первой букве и выводит группы в алфавитном порядке букв. Внутри каждой группы слова выводятся в алфавитном порядке через пробел. Формат: буква: слово1 слово2 .... Если на какую-то букву нет слов, она не выводится.
Входные данные:
Произвольное количество строк со словами.
Выходные данные:
Строки в формате буква: слова.
Пример ввода:
кот дом мост
стол лампа
парк торт
Пример вывода:
к: парк
м: дом лампа
п: лампа
т: кот мост стол торт
Возможное решение:

```
import sys
groups = {}
for line in sys.stdin:
    for word in line.split():
        first = word[0]
        groups.setdefault(first, []).append(word)
for letter in sorted(groups):
    print(f'{letter}: {" ".join(sorted(groups[letter]))}')
```

Задание 5 (написание кода)
Напиши программу, которая с помощью регулярных выражений находит в тексте все даты в формате ДД.ММ.ГГГГ (день и месяц могут быть однозначными). Выведи найденные даты через пробел в порядке появления. Если дат нет, программа ничего не выводит.
Входные данные:
Одна строка с текстом.
Выходные данные:
Строка с датами (через пробел).
Пример ввода:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Событие произошло 4.10.1957, а затем 03.11.1957.
Пример вывода:
4.10.1957 03.11.1957
Возможное решение:

```
import re
text = input()
pattern = r'\d{1,2}\.\d{1,2}\.\d{4}'
dates = re.findall(pattern, text)
print(' '.join(dates))
```

Задание 6 (написание кода)
Напиши программу calc.py, которая принимает два целых числа (позиционные аргументы) и опциональный ключ -o (или --op) с возможными значениями: add, sub, mul, div. Значение по умолчанию – add. Программа должна вывести результат операции. В случае ошибки (например, деление на ноль) вывести строку «Error». Используй модуль argparse.
Примеры вызова:

```
$ python calc.py 5 3
8
$ python calc.py 5 3 -o mul
15
$ python calc.py 5 0 -o div
Error
```

Входные данные:
Аргументы командной строки.
Выходные данные:
Результат операции или «Error».
Возможное решение:

```
import argparse
ops = {
    'add': lambda a, b: a + b,
    'sub': lambda a, b: a - b,
    'mul': lambda a, b: a * b,
    'div': lambda a, b: a / b
}
parser = argparse.ArgumentParser()
parser.add_argument('a', type=int)
parser.add_argument('b', type=int)
parser.add_argument('-o', '--op', choices=ops.keys(), default='add')
args = parser.parse_args()
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

```
try:
    print(ops[args.op](args.a, args.b))
except ZeroDivisionError:
    print('Error')
```

Задание 7 (написание кода)
Напиши программу, которая парсит страницу https://docs.python.org/3/library/functions.html и выводит все ссылки на документацию функций, названия которых начинаются с буквы a (регистр не учитывается). Выведи только ссылки (без названий), каждая на новой строке. Используй requests и BeautifulSoup.
Входные данные:
Нет.
Выходные данные:
Ссылки на документацию, начинающиеся с a.
Пример вывода:
https://docs.python.org/3/library/functions.html#abs
https://docs.python.org/3/library/functions.html#aiter
...
Возможное решение:

```
import requests
from bs4 import BeautifulSoup
url = 'https://docs.python.org/3/library/functions.html'
response = requests.get(url)
soup = BeautifulSoup(response.text, 'html.parser')
for div in soup.find_all('div', class_='line'):
    a = div.find('a')
    if a and a.text.startswith('a'):
        print(f'{url}{a["href"]}')
```

Задание 8 (написание кода)
Напиши программу, которая по введённой дате (в формате ДД.ММ.ГГГГ) определяет день недели и выводит его название на русском языке (Понедельник, Вторник, … Воскресенье). Используй модуль datetime.
Входные данные:
Одна строка с датой.
Выходные данные:
Название дня недели.
Пример ввода:
01.01.2024
Пример вывода:

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Понедельник
Возможное решение:

```
import datetime
date_str = input()
date = datetime.datetime.strptime(date_str, '%d.%m.%Y')
weekdays = ['Понедельник', 'Вторник', 'Среда', 'Четверг', 'Пятница', 'Суббота', 'Воскресенье']
print(weekdays[date.weekday()])
```

Задание 9 (исправление ошибок)
Дана программа, которая должна читать JSON-файл и выводить значение ключа value. Код содержит ошибку – используется неправильный метод чтения. Исправь.

```
import json
with open('data.json', encoding='utf-8') as f:
    data = f.read()
print(data['value'])
```

Входные данные:
Файл data.json с корректным JSON.
Выходные данные:
Значение ключа value.
Пример:
Если файл содержит {"value": 42}, программа должна вывести 42.
Возможное решение:

```
import json
with open('data.json', encoding='utf-8') as f:
    data = json.load(f)   # исправлено: использовать json.load вместо f.read()
print(data['value'])
```

Задание 10 (исправление ошибок)
Программа должна проверять, является ли строка правильной скобочной последовательностью (только круглые скобки). Код содержит ошибку – не проверяется случай, когда стек пуст, а встретилась закрывающая скобка. Исправь.

```
s = input()
stack = []
for ch in s:
    if ch == '(':
        stack.append(ch)
    elif ch == ')':
        stack.pop()
```

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

```
if stack:
    print(False)
else:
    print(True)
```
Входные данные:
Строка из скобок.
Выходные данные:
True или False.
Пример ввода:
```
())(
```
Пример вывода:
```
False
```
Возможное решение:
```
s = input()
stack = []
for ch in s:
    if ch == '(':
        stack.append(ch)
    elif ch == ')':
        if not stack:   # добавлена проверка на пустоту
            print(False)
            break
        stack.pop()
else:
    print(len(stack) == 0)
```

Задание 11 (исправление ошибок)
Программа должна вычислять сумму чисел из файла, но код содержит ошибку – не учитывается, что числа могут быть в нескольких строках. Исправь.
```
with open('data.txt') as f:
    line = f.readline()
    total = sum(map(int, line.split()))
print(total)
```
Входные данные:
Файл data.txt с числами в нескольких строках.
Выходные данные:
Сумма всех чисел.

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

	Пример файла data.txt:
	10 20
	30 40
	Ожидаемый вывод:
	100.
	Возможное решение:
	total = 0
	with open('data.txt') as f:
	  for line in f:
	  total += sum(map(int, line.split()))
	print(total)
	
	Задание 12 (исправление ошибок)
	Программа должна искать все email-адреса в тексте, но регулярное выражение написано с ошибкой (не учитывает точки в домене). Исправь.
	import re
	text = input()
	emails = re.findall(r'\w+@\w+', text)
	print(' '.join(emails))
	Входные данные:
	Строка с email-адресами.
	Пример ввода:
	user@example.com и test@mail.ru.
	Ожидаемый вывод:
	user@example.com test@mail.ru.
	Возможное решение:
	import re
	text = input()
	emails = re.findall(r'[\w.-]+@[\w.-]+', text) # исправлено
	print(' '.join(emails))
	
	Задание 13 (краткий ответ)
	Вопрос: Какой метод модуля json используется для записи объекта Python в файл в формате JSON? Запиши название метода.
	Ответ: dump

	Задание 14 (краткий ответ) Вопрос: Какой класс модуля csv позволяет читать CSV-файл, преобразуя каждую строку в словарь, где ключи – это заголовки столбцов? Запиши название класса. Ответ: DictReader Задание 15 (краткий ответ) Вопрос: Какой модуль Python используется для работы с регулярными выражениями? Запиши название модуля. Ответ: re
Шкала оценивания, нижнее значение	0
Шкала оценивания, верхнее значение	15
Шкала оценивания, минимальный проходной балл	9

Итоговый контроль/аттестация по ДОП	
Количество академических часов	2.00
Формы контроля	Защита итогового проекта, разработанного в рамках выполнения кейс-задачи, включает демонстрацию полученного результата и ответы на вопросы согласно утвержденному плану: Обоснование выбранной проблемы. Цель проекта и ожидаемый результат. Предложенное решение. Распределение задач между участниками. Демонстрация результата и его работоспособности. Анализ сильных сторон, недостатков и ограничений решения. Перспективы развития проекта.
Диагностические инструменты	Экспертный лист оценки проекта по установленным критериям Материалы итоговой защиты проекта

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Маршеня Виктория Вадимовна
серийный номер 01ba2191c94f2eb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Показатели и критерии оценивания	Проект оценивается по следующим критериям: соответствие заданию; проработка проекта; оригинальность решения; техническая сложность; практическая применимость; командная работа; защита (презентация) проекта. Каждый критерий оценивается по шкале от 0 до 3 баллов. Максимальное количество баллов — 21. Итоговая аттестация считается пройденной при наборе не менее 13 баллов (60% от максимального количества баллов). Подробное описание критериев, уровней оценивания и порядка выставления баллов представлено в Экспертном листе оценки проекта.
Варианты заданий	Защита итогового проекта, разработанного в ходе выполнения кейс-задачи, осуществляется в соответствии с требованиями, представленными в приложении к дополнительной образовательной программе (ДОП) в составе учебно-методического комплекса (УМК).
Шкала оценивания, нижнее значение	0.00
Шкала оценивания, верхнее значение	21.00
Шкала оценивания, минимальный проходной балл	13.00

Учебно-методические материалы

Наименование поля	Значение полей	Значение полей	Значение полей	Значение полей
Порядковый номер модуля	1	2	3	4
Методы, формы и технологии	Обучение реализуется на основе деятельностного, компетентностного и практико-ориентированного подходов. Освоение содержания модуля осуществляется через выполнение практических заданий, решение алгоритмических задач, разработку программ и реализацию проектов. Значительная часть учебного времени отводится практической деятельности, направленной на формирование навыков программирования и применение полученных знаний для решения практических задач. В образовательном процессе используются объяснение нового материала, демонстрация примеров, практические работы, решение задач, проектная деятельность, консультации, самостоятельная работа обучающихся и разбор типовых ошибок. Обучение осуществляется в групповой форме под руководством преподавателя. В зависимости от вида учебной деятельности используются фронтальная, индивидуальная и групповая формы организации работы обучающихся. В качестве образовательных технологий используются электронное обучение, дистанционные образовательные технологии, проектное обучение, технологии применения инструментов искусственного интеллекта, автоматизированная проверка заданий, средства визуализации алгоритмов и программного кода, а также современные цифровые инструменты разработки программ.	Обучение реализуется на основе деятельностного, компетентностного и практико-ориентированного подходов. Освоение содержания модуля осуществляется через выполнение практических заданий, решение алгоритмических задач, разработку программ и реализацию проектов. Значительная часть учебного времени отводится практической деятельности, направленной на формирование навыков программирования и применение полученных знаний для решения практических задач. В образовательном процессе используются объяснение нового материала, демонстрация примеров, практические работы, решение задач, проектная деятельность, консультации, самостоятельная работа обучающихся и разбор типовых ошибок. Обучение осуществляется в групповой форме под руководством преподавателя. В зависимости от вида учебной деятельности используются фронтальная, индивидуальная и групповая формы организации работы обучающихся. В качестве образовательных технологий используются электронное обучение, дистанционные образовательные технологии, проектное обучение, технологии применения инструментов искусственного интеллекта, автоматизированная проверка заданий, средства визуализации алгоритмов и программного кода, а также современные цифровые инструменты разработки программ.	Обучение реализуется на основе деятельностного, компетентностного и практико-ориентированного подходов. Освоение содержания модуля осуществляется через выполнение практических заданий, решение алгоритмических задач, разработку программ и реализацию проектов. Значительная часть учебного времени отводится практической деятельности, направленной на формирование навыков программирования и применение полученных знаний для решения практических задач. В образовательном процессе используются объяснение нового материала, демонстрация примеров, практические работы, решение задач, проектная деятельность, консультации, самостоятельная работа обучающихся и разбор типовых ошибок. Обучение осуществляется в групповой форме под руководством преподавателя. В зависимости от вида учебной деятельности используются фронтальная, индивидуальная и групповая формы организации работы обучающихся. В качестве образовательных технологий используются электронное обучение, дистанционные образовательные технологии, проектное обучение, технологии применения инструментов искусственного интеллекта, автоматизированная проверка заданий, средства визуализации алгоритмов и программного кода, а также современные цифровые инструменты разработки программ.	Обучение реализуется на основе деятельностного, компетентностного и практико-ориентированного подходов. Освоение содержания модуля осуществляется через выполнение практических заданий, решение алгоритмических задач, разработку программ и реализацию проектов. Значительная часть учебного времени отводится практической деятельности, направленной на формирование навыков программирования и применение полученных знаний для решения практических задач. В образовательном процессе используются объяснение нового материала, демонстрация примеров, практические работы, решение задач, проектная деятельность, консультации, самостоятельная работа обучающихся и разбор типовых ошибок. Обучение осуществляется в групповой форме под руководством преподавателя. В зависимости от вида учебной деятельности используются фронтальная, индивидуальная и групповая формы организации работы обучающихся. В качестве образовательных технологий используются электронное обучение, дистанционные образовательные технологии, проектное обучение, технологии применения инструментов искусственного интеллекта, автоматизированная проверка заданий, средства визуализации алгоритмов и программного кода, а также современные цифровые инструменты разработки программ.
Методические разработки (представлены в качестве приложения к ДОП)	Методические разработки модуля представлены в виде приложений к ДОП и включают описание модуля, поурочные планы учебных занятий, методические рекомендации по организации и проведению практических занятий, индивидуальных и командных проектов, самостоятельной работы обучающихся, текущего контроля и промежуточной аттестации. В состав методических разработок входят рекомендации по проведению занятий, краткое изложение теоретического материала, методические указания по выполнению заданий, формы контроля, показатели и критерии оценивания, возможные решения, а также инструкции по организации образовательного процесса, работе с образовательной платформой и системой автоматической проверки заданий.	Методические разработки модуля представлены в виде приложений к ДОП и включают описание модуля, поурочные планы учебных занятий, методические рекомендации по организации и проведению практических занятий, индивидуальных и командных проектов, самостоятельной работы обучающихся, текущего контроля и промежуточной аттестации. В состав методических разработок входят рекомендации по проведению занятий, краткое изложение теоретического материала, методические указания по выполнению заданий, формы контроля, показатели и критерии оценивания, возможные решения, а также инструкции по организации образовательного процесса, работе с образовательной платформой и системой автоматической проверки заданий.	Методические разработки модуля представлены в виде приложений к ДОП и включают описание модуля, поурочные планы учебных занятий, методические рекомендации по организации и проведению практических занятий, индивидуальных и командных проектов, самостоятельной работы обучающихся, текущего контроля и промежуточной аттестации. В состав методических разработок входят рекомендации по проведению занятий, краткое изложение теоретического материала, методические указания по выполнению заданий, формы контроля, показатели и критерии оценивания, возможные решения, а также инструкции по организации образовательного процесса, работе с образовательной платформой и системой автоматической проверки заданий.	Методические разработки модуля представлены в виде приложений к ДОП и включают описание модуля, поурочные планы учебных занятий, методические рекомендации по организации и проведению практических занятий, индивидуальных и командных проектов, самостоятельной работы обучающихся, текущего контроля, промежуточной и итоговой аттестации, а также методические рекомендации по организации и проведению кейс-задачи. В состав методических разработок входят рекомендации по проведению занятий, краткое изложение теоретического материала, методические указания по выполнению заданий, формы контроля, показатели и критерии оценивания, возможные решения, материалы итоговой аттестации, а также инструкции по организации образовательного процесса, работе с образовательной платформой и системой автоматической проверки заданий.
Материалы модуля (представлены в качестве приложения к ДОП)	Материалы модуля представлены в виде приложений к ДОП и включают презентационные материалы, материалы практических занятий, индивидуальных и командных проектов, самостоятельной работы обучающихся. В состав материалов входят описания заданий, необходимые учебные и справочные материалы, а также дополнительные материалы и файлы, используемые при освоении содержания модуля.	Материалы модуля представлены в виде приложений к ДОП и включают презентационные материалы, материалы практических занятий, индивидуальных и командных проектов, самостоятельной работы обучающихся. В состав материалов входят описания заданий, необходимые учебные и справочные материалы, а также дополнительные материалы и файлы, используемые при освоении содержания модуля.	Материалы модуля представлены в виде приложений к ДОП и включают презентационные материалы, материалы практических занятий, индивидуальных и командных проектов, самостоятельной работы обучающихся. В состав материалов входят описания заданий, необходимые учебные и справочные материалы, а также дополнительные материалы и файлы, используемые при освоении содержания модуля.	Материалы модуля представлены в виде приложений к ДОП и включают презентационные материалы, материалы практических занятий, индивидуальных и командных проектов, самостоятельной работы обучающихся, также материалы кейс-задачи. В состав материалов входят описания заданий, необходимые учебные и справочные материалы, а также дополнительные материалы и файлы, используемые при освоении содержания модуля.

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Марценя Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Учебная литература	Мэтиз, Э. Изучаем Python: программирование игр, визуализация данных, веб-приложения. 3-е изд. / Эрик Мэтиз; пер. с англ. Е. П. Матвеев; ил. Л. Ю. Егорова. — Москва : Питер, 2022. — 512 с. Васильев, А. Н. Программирование на Python в примерах и задачах / Алексей Николаевич Васильев. — Москва : Бомбора, 2023. — 616 с. Любанович, Б. Простой Python. Современный стиль программирования. 2-е изд. / Билл Любанович; пер. с англ. И. А. Пальти, Е. В. Зазноба; ил. Н. Гринчик. — Москва : Питер, 2022. — 592 с.	Мэтиз, Э. Изучаем Python: программирование игр, визуализация данных, веб-приложения. 3-е изд. / Эрик Мэтиз; пер. с англ. Е. П. Матвеев; ил. Л. Ю. Егорова. — Москва : Питер, 2022. — 512 с. Васильев, А. Н. Программирование на Python в примерах и задачах / Алексей Николаевич Васильев. — Москва : Бомбора, 2023. — 616 с. Любанович, Б. Простой Python. Современный стиль программирования. 2-е изд. / Билл Любанович; пер. с англ. И. А. Пальти, Е. В. Зазноба; ил. Н. Гринчик. — Москва : Питер, 2022. — 592 с.	Мэтиз, Э. Изучаем Python: программирование игр, визуализация данных, веб-приложения. 3-е изд. / Эрик Мэтиз; пер. с англ. Е. П. Матвеев; ил. Л. Ю. Егорова. — Москва : Питер, 2022. — 512 с. Васильев, А. Н. Программирование на Python в примерах и задачах / Алексей Николаевич Васильев. — Москва : Бомбора, 2023. — 616 с. Любанович, Б. Простой Python. Современный стиль программирования. 2-е изд. / Билл Любанович; пер. с англ. И. А. Пальти, Е. В. Зазноба; ил. Н. Гринчик. — Москва : Питер, 2022. — 592 с. Бхаргава, А. Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих / Адитья Бхаргава; пер. с англ. Е. Матвеев; ред. Н. Римицан; ил. С. Маликова. — Москва : Питер, 2022. — 288 с. — (Библиотека программиста). — ISBN 978-5-4461-0923-4 Прохоренок, Н. А. Python 3 и PyQt 5. Разработка приложений. — 2-е изд., перераб. и доп. / Н. А. Прохоренок, В. А. Дронов. — СПб.: БХВ-Петербург, 2018	Мэтиз, Э. Изучаем Python: программирование игр, визуализация данных, веб-приложения. 3-е изд. / Эрик Мэтиз; пер. с англ. Е. П. Матвеев; ил. Л. Ю. Егорова. — Москва : Питер, 2022. — 512 с. Васильев, А. Н. Программирование на Python в примерах и задачах / Алексей Николаевич Васильев. — Москва : Бомбора, 2023. — 616 с. Любанович, Б. Простой Python. Современный стиль программирования. 2-е изд. / Билл Любанович; пер. с англ. И. А. Пальти, Е. В. Зазноба; ил. Н. Гринчик. — Москва : Питер, 2022. — 592 с.

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Политика использования инструментов искусственного интеллекта Получателями поддержки при освоении дополнительной общеобразовательной программы

(входит в состав Учебно-методического комплекса)

1. Общие положения

1.1. Настоящая Политика устанавливает единые требования к использованию инструментов искусственного интеллекта (далее – ИИ-инструменты) Получателями поддержки при освоении дополнительной общеобразовательной программы (далее – Программа) и является локальным нормативным актом Исполнителя, обязательным для исполнения в рамках Договора оказания платных образовательных услуг (далее – Договор).

1.2. Политика входит в состав Учебно-методического комплекса Программы и размещается на Образовательной платформе в открытом доступе.

1.3. Ознакомление с Политикой и принятие обязательств по ее соблюдению является неотъемлемым условием зачисления Получателя поддержки на Программу.

1.4. Политика разработана в целях обеспечения объективности оценки результатов обучения, формирования у Получателей поддержки навыков критического мышления и академической честности, защиты интеллектуальной собственности и персональных данных, предотвращения нарушений, связанных с недобросовестным использованием ИИ-инструментов.

1.5. Действие Политики распространяется на все виды учебной деятельности, предусмотренные учебным планом Программы, включая теоретическую подготовку, практические работы, выполнение заданий текущего контроля, промежуточной и итоговой аттестации, а также Итоговое тестирование и социологическое исследование (ИТиСИ).

2. Термины и определения

2.1. ИИ-инструменты – программное обеспечение, веб-сервисы, мобильные приложения и иные цифровые инструменты, функционирующие на основе технологий искусственного интеллекта (включая генеративные нейросети, интеллектуальные помощники для написания кода, алгоритмы машинного обучения).

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

2.2. Получатель поддержки – физическое лицо, зачисленное на обучение по Программе в соответствии с Договором.

2.3. Исполнитель – Общество с ограниченной ответственностью «Учи.Дома».

2.4. Образовательная платформа – платформа, принадлежащая Исполнителю, расположенная по адресу https://app.doma.uchi.ru/, с использованием которой осуществляется обучение по Программе.

2.5. Кейс-задача – практико-ориентированное задание, не имеющее единственного верного ответа, предполагающее поиск и предложение наиболее оптимального и аргументированного решения.

2.6. ИТиСИ – Итоговое тестирование и социологическое исследование, проводимое в целях независимой оценки результатов обучения.

2.7. Договор – Договор оказания платных образовательных услуг, заключенный между Исполнителем и законным представителем Получателя поддержки или Получателем поддержки в рамках Мероприятия.

2.8. Мероприятие – мероприятие, направленное на реализацию дополнительных образовательных программ, разработанных для обучения школьников 8-11 классов и обучающихся по программам среднего профессионального образования по дополнительному курсу по современным языкам программирования, технологиям искусственного интеллекта и робототехнике, в части достижения результата предоставления субсидии.

2.9. Федеральный оператор – уполномоченная организация, выполняющая функции по организационно-финансовому, информационному и технологическому обеспечению функционирования системы полной или частичной компенсации затрат на обучение по дополнительным образовательным программам – Автономная некоммерческая организация «Университет национальной технологической инициативы 2035».

3. Общие принципы использования ИИ-инструментов

3.1. Использование ИИ-инструментов допускается исключительно как вспомогательный инструмент для повышения эффективности обучения. Оно не может подменять собой самостоятельное формирование у Получателя поддержки знаний, умений и навыков, предусмотренных Программой.

3.2. Получатель поддержки несет полную ответственность за достоверность, корректность и оригинальность представленных к оценке результатов учебной деятельности, вне зависимости от того, использовались ли ИИ-инструменты при их подготовке.

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

3.3. Использование ИИ-инструментов не освобождает Получателя поддержки от необходимости критической проверки и верификации полученных результатов. Получатель поддержки обязан проверить сгенерированный код, факты, ссылки и любые иные выходные данные на предмет ошибок, неточностей и вымышленных фактов или несуществующих ссылок.

3.4. Получатель поддержки должен осознавать, что ИИ-инструменты не создают новых знаний, а лишь компилируют и перерабатывают существующие данные на основе статистических алгоритмов. ИИ не может заменить самостоятельное мышление и понимание предмета.

3.5. При использовании ИИ-инструментов Получатель поддержки обязан соблюдать требования законодательства о персональных данных и не загружать в них персональные данные, коммерческую тайну или иную конфиденциальную информацию.

4. Перечень видов учебной деятельности, при выполнении которых использование ИИ-инструментов допускается

4.1. Использование ИИ-инструментов допускается при выполнении следующих видов учебной деятельности:

4.1.1. Теоретическая подготовка – для переформулирования, структурирования и углубленного объяснения теоретического материала. Получатель поддержки обязан проверить полученные разъяснения по официальным источникам.

4.1.2. Поиск и резюмирование информации – для быстрого поиска и резюмирования общедоступной информации, а также для поиска дополнительных источников. Получатель поддержки обязан указать ссылки на все первоисточники.

4.1.3. Освоение примеров и синтаксиса – для уточнения примеров на нейтральных данных, не связанных с решаемой учебной задачей.

4.1.4. Отладка и рефакторинг собственных разработок – для поиска ошибок в собственном коде и получения рекомендаций по их исправлению. Запрещается использовать ИИ для генерации кода вместо собственного неработающего решения.

4.1.5. Работа с кодом, сгенерированным ИИ – для критического анализа, проверки, исправления и улучшения кода, созданного ИИ-инструментом.

5. Перечень видов учебной деятельности, при выполнении которых использование ИИ-инструментов запрещено

5.1. Использование ИИ-инструментов категорически запрещено при выполнении следующих видов учебной деятельности:

Контур.Крипто
владелец Маршеня Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

5.1.1. Решение задач текущего контроля.

5.1.2. Прохождение промежуточной аттестации по итогам модулей.

5.1.3. Прохождение итоговой аттестации по Программе (включая защиту проекта, итоговое тестирование).

5.1.4. Прохождение Итогового тестирования и социологического исследования (ИТиСИ). Нарушение данного запрета является грубым нарушением условий участия в Мероприятии.

5.1.5. Формирование готового решения Кейс-задачи.

5.1.6. Формирование готовых решений для выполнения практических заданий (кода, скрипта, модели, конфигурации).

5.1.7. Загрузка персональных данных или иной конфиденциальной информации, а также использование ИИ-инструментов для создания контента, который может нарушать исключительные права третьих лиц.

5.1.8. Использование ИИ для подмены личности или обхода процедур идентификации, включая ИТиСИ.

5.1.9. Поиск готовых решений при участии в соревновательных форматах обучения и иных дополнительных мероприятиях (в том числе мероприятиях экосистемы обучения), если использование ИИ-инструментов прямо не разрешено условиями проведения такого мероприятия.

6. Последствия нарушения Политики

6.1. Обнаружение факта использования ИИ-инструментов в видах деятельности, указанных в разделе 5, является основанием для: аннулирования результата работы с выставлением оценки «неудовлетворительно» (0 баллов); недопуска Получателя поддержки к следующему этапу аттестации (в случае нарушения при промежуточной или итоговой аттестации).

6.2. В случае выявления факта использования ИИ-инструментов при прохождении ИТиСИ, информация незамедлительно направляется Федеральному оператору.

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

6.3. Повторное (более 2 раз) нарушение Политики в течение одного учебного года является основанием для рассмотрения вопроса об отчислении Получателя поддержки в порядке, установленном Договором и законодательством Российской Федерации.

7. Порядок действий преподавателя при выявлении нарушения

7.1. При обнаружении признаков, указывающих на возможное использование ИИ-инструментов (в том числе на основании уведомлений системы автоматизированного контроля), преподаватель фиксирует эти признаки.

7.2. Выявленные признаки являются техническими индикаторами, требующими последующего анализа, и не являются автоматическим доказательством нарушения.

7.3. При наличии достаточных оснований преподаватель направляет Получателю поддержки запрос о предоставлении письменного или устного пояснения по логике выполнения работы. Получатель поддержки обязан предоставить запрашиваемое пояснение в установленный преподавателем срок (не позднее 1 рабочего дня).

8. Заключительные положения

8.1. Настоящая Политика является неотъемлемой частью УМК Программы и локальным нормативным актом Исполнителя. Политика вступает в силу с момента начала реализации Программы.

8.2. Информирование Получателей поддержки о положениях Политики осуществляется на первом занятии, а также путем размещения текста Политики на Образовательной платформе.

8.3. Изменения в Политику вносятся в порядке, установленном для внесения изменений в УМК Программы, и доводятся до сведения Получателя поддержки через Образовательную платформу не позднее чем за 5 рабочих дней до даты вступления изменений в силу.

8.4. Во всем, что не урегулировано настоящей Политикой, применяются положения Договора и действующего законодательства Российской Федерации.

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027