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 (Обучающиеся СПО) Онлайн
3.	Название каждой ДОП Комплекта ДОП	Python Старт (базовый уровень, СПО) Python Про (продвинутый уровень, СПО)
4.	Вид ДОП входящих в Комплект ДОП	Дополнительные общеобразовательные программы
5.	Ссылка на Курс	https://futurecode.uchi.ru/python-spo-online/
6.	Форма обучения	Очная
7.	Формат обучения	Онлайн
8.	Целевая аудитория ДОП	Обучающиеся по программам СПО
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, СПО)» и «Python Про (продвинутый уровень, СПО)» имеют техническую направленность и относятся к направлению обучения «Современные языки программирования». Программа ориентирована на студентов 1–4 курсов организаций среднего профессионального образования и разработана с учетом их возрастных особенностей, уровня базовой подготовки в области информатики, познавательных интересов и образовательных потребностей. Содержание программы направлено на системное формирование алгоритмического мышления, цифровой грамотности, навыков программирования и решения прикладных задач средствами языка Python, а также на подготовку к профессиональной деятельности в сфере информационных технологий. 2) краткое описание цели и содержания каждой ДОП Python Старт (базовый уровень, СПО) Цель программы — формирование у обучающихся по программам среднего профессионального образования специализированных знаний и практических навыков программирования на языке Python, развитие алгоритмического мышления, профессиональных цифровых компетенций, навыков ответственного использования ИИ-технологий и подготовка к профессиональной деятельности в сфере информационных технологий через решение практических задач и выполнение проектов. Программа направлена на изучение языка программирования Python, объектно-ориентированного программирования, современных технологий разработки программного обеспечения и работы с данными, выполнение практических работ по созданию алгоритмов и решению классических задач, а также развитие навыков проектной деятельности посредством выполнения индивидуальных и командных проектов, формирование практического опыта разработки прикладных программных решений с учетом возрастных особенностей обучающихся, их уровня подготовки и образовательных потребностей. Содержание: Изучение базовых конструкций языка Python: переменных, типов данных, операторов, условий, циклов, строк, коллекций, функций и модулей. Освоение основ алгоритмизации и программирования через решение типовых задач. Выполнение практических работ по созданию консольных программ и обработке данных. Изучение основ объектно-ориентированного программирования: классов и объектов, методов классов, базовых принципов наследования, полиморфизма и инкапсуляции. Работа с файлами и форматами данных JSON и CSV. Освоение основ работы с внешними инструментами и библиотеками:
		командная строка, работа с датой и временем, регулярные выражения, тестирование и логирование программ. Изучение основ веб-взаимодействия: API, парсинг данных и основы работы с системами контроля версий. Изучение основ разработки графических интерфейсов пользователя. Работа с базами данных: основы SQL и выполнение простых запросов. Использование ИИ-инструментов в программировании. Выполнение учебных проектов для закрепления изученного материала, а также индивидуальных и командных учебных проектов. Решение кейс-задачи — выполнение практического задания по решению проблемной ситуации в рамках направления обучения. Python Про (продвинутый уровень, СПО) Цель программы — формирование у обучающихся по программам среднего профессионального образования углубленных знаний и практических навыков программирования на языке Python, совершенствование алгоритмического мышления, развитие профессиональных цифровых компетенций, навыков ответственного использования ИИ-технологий и подготовка к профессиональной деятельности 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, веб-разработки и работы с данными, развитие навыков проектной деятельности посредством выполнения индивидуальных и командных проектов, формирование практического опыта разработки прикладных программных решений и полноценных программных продуктов, 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, областей видимости и принципов организации программ. Изучение и практическое применение алгоритмов поиска и сортировки, а также оценки их сложности. Работа с файлами, форматами JSON и CSV, регулярными выражениями, тестированием и логированием программ. Освоение работы с API и парсинга данных, а также систем контроля версий. Изучение разработки графических интерфейсов с использованием PyQt. Работа с базами данных: SQL и выполнение запросов, взаимодействие Python с базами данных. Изучение веб-разработки с использованием Flask: создание веб-приложений,

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

		маршрутизация, шаблонизация и обработка форм. Работа с SQLAlchemy и основами ORM. Изучение структуры веб-приложений и развертывания проектов. Решение задач повышенной сложности, направленных на развитие алгоритмического мышления и навыков разработки реальных программных продуктов. 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, СПО) Специализированные знания и навыки программирования не требуются. Требуется наличие знаний по математике и информатике на уровне не ниже 9 класса, навыков работы с персональным компьютером, логического мышления, общей цифровой грамотности и знание базовой терминологии в области алгоритмизации и программирования. Программа ориентирована на формирование практических навыков программирования и компетенций, необходимых для решения типовых профессионально-ориентированных задач, выполнение практических работ по созданию алгоритмов и решению классических задач программирования, а также разработку учебного проекта. Python Про (продвинутый уровень, СПО) Требуются специализированные знания и навыки программирования на языке Python. Предполагается наличие знаний по математике и информатике на уровне не ниже 9 класса, понимание основных конструкций языка Python, опыт решения практических задач, развитое логическое мышление, навыки самостоятельной работы с учебными материалами и общая цифровая грамотность. Программа ориентирована на формирование профессиональных компетенций в области разработки программного обеспечения, углубленное изучение современных технологий разработки программного обеспечения, 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, СПО) 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, освоят основные принципы и парадигму объектно-ориентированного программирования, включая использование классов и объектов, методов классов,
		принципов инкапсуляции, наследования и полиморфизма, 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, типов данных и базовых возможностей языка программирования Python, освоение основ объектно-ориентированного программирования, 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. Python Про (продвинутый уровень, СПО) 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, освоят разработку графических интерфейсов с использованием библиотеки PyQt, работу с базами данных и языком SQL, а также разработку веб-приложений с использованием микрофреймворка Flask, шаблонов Jinja, SQLAlchemy и принципов ORM, 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, веб-разработки и работы с данными, развитие навыков проектной деятельности посредством выполнения индивидуальных и командных проектов, формирование практического опыта разработки прикладных программных решений и полноценных программных продуктов, а также подготовку к участию в олимпиадах, конкурсах и соревнованиях в сфере информационных технологий с учетом уровня подготовки и образовательных потребностей обучающихся.

Основные данные ДОП: БАЗОВЫЙ УРОВЕНЬ

№ п/п	Наименование поля	Значение поля
1.	Название ДОП	Python Старт (базовый уровень, СПО)
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 01ba3191e94f2eb04e6f27ee019aedad
срок действия 05.12.2025 - 05.03.2027

7.	Кейс-задача	Название Кейс-задачи: Подготовка бейджей для мероприятий Описание проблемной ситуации: При проведении конференций, форумов, хакатонов, соревнований и других мероприятий организаторам необходимо заранее подготовить бейджи для участников, волонтёров, экспертов и гостей. Если бейджи создаются вручную, приходится многократно вводить одни и те же данные, копировать информацию из таблиц, размещать текст на макете и следить за правильностью оформления. При большом количестве участников такая работа занимает много времени и повышает риск ошибок в именах, ролях, названиях организаций и ссылках. Дополнительно на бейджах часто требуется размещать QR-коды, например со ссылкой на профиль участника, страницу мероприятия, расписание или форму обратной связи. Цель решения: Разработать командное решение, которое поможет организаторам быстро создавать бейджи с индивидуальными данными участников и QR-кодами. Команда самостоятельно определяет способ достижения результата, формат входных данных, внешний вид бейджа и состав разрабатываемого продукта. Контекст: Бейджи используются на образовательных, деловых, спортивных и культурных мероприятиях для быстрой идентификации участников. Обычно на бейдже размещают имя, фамилию, роль, организацию, название мероприятия и дополнительную информацию для перехода по QR-коду. Современные мероприятия могут насчитывать десятки, сотни и даже тысячи участников, поэтому подготовка бейджей является важной частью работы организаторов. При этом необходимо обеспечить единый стиль оформления, корректность данных и возможность быстро подготовить материалы для печати или повторной генерации при изменении информации об участниках. 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: 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 для работы с PDF-файлами, изображениями, QR-кодами, таблицами, базами данных и другими форматами данных. 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, изучение языка программирования Python, объектно-ориентированного программирования, современных технологий разработки программного обеспечения и работы с данными, выполнение практических работ по созданию алгоритмов и решению классических задач, а также развитие навыков проектной деятельности посредством выполнения индивидуальных и командных проектов, формирование практического опыта разработки прикладных программных решений 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, профессиональных цифровых компетенций, навыков ответственного использования ИИ-технологий и подготовка к профессиональной деятельности в сфере информационных технологий.
2.	Цель и задачи ДОП	Цель программы: Формирование у обучающихся по программам среднего профессионального образования специализированных знаний и практических навыков программирования на языке Python, развитие алгоритмического мышления, профессиональных цифровых компетенций, навыков ответственного использования ИИ-технологий и подготовка к профессиональной деятельности в сфере информационных технологий через решение практических задач и выполнение 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; сформировать знания основ объектно-ориентированного программирования, включая использование классов и объектов, методов классов, принципов инкапсуляции, наследования и полиморфизма; обучить применению современных технологий разработки программного обеспечения, включая работу с файлами, API, форматами JSON и CSV, регулярными выражениями, тестированием и системами контроля версий; сформировать навыки разработки графических интерфейсов с использованием библиотеки PyQt, а также работы с базами данных, включая основы языка SQL, выполнение запросов и взаимодействие с базами данных средствами Python; 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 подготовке к профессиональной деятельности 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 программного обеспечения; о принципах объектно-ориентированного программирования; о современных технологиях разработки программного обеспечения; о принципах разработки графических интерфейсов и работы с базами данных; о возможностях использования технологий искусственного интеллекта в процессе разработки программ; о принципах ответственного и безопасного использования информационных технологий; о возможностях профессиональной деятельности в сфере информационных технологий. Знать: основные конструкции языка Python: переменные, типы данных, операторы, условные конструкции, циклы, строки, коллекции, функции и модули; основы объектно-ориентированного программирования, включая использование классов и объектов, методов классов, принципов инкапсуляции, наследования и полиморфизма; современные технологии разработки программного обеспечения, включая работу с файлами, API, форматами JSON и CSV, регулярными выражениями, тестированием и системами контроля версий; основы разработки графических интерфейсов с использованием библиотеки PyQt; основы языка SQL, принципы выполнения запросов и взаимодействия с базами данных средствами Python; 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; принципы разработки графических интерфейсов и взаимодействия с базами 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; использовать основные конструкции языка Python при решении практических задач; применять принципы объектно-ориентированного программирования 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; разрабатывать графические интерфейсы с использованием библиотеки PyQt; работать с базами данных, использовать язык SQL, выполнять запросы и взаимодействовать с базами данных средствами Python; 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 разработки графических интерфейсов и работы с базами данных; навыками поиска, анализа и исправления ошибок в программном коде; навыками ответственного использования технологий искусственного интеллекта; навыками

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

		проектной, исследовательской и командной деятельности при выполнении индивидуальных и командных проектов. Формируемые компетенции Программа ориентирована на формирование следующих компетенций в соответствии с ФГОС СПО по специальности 09.02.07 Информационные системы и программирование: Общие компетенции ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам. ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности. ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие. ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами. ОК 09. Использовать информационные технологии в профессиональной деятельности. ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языках. Профессиональные компетенции ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием. ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием. ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств. ПК 1.4. Выполнять тестирование программных модулей. ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода. ПК 5.4. Производить разработку модулей информационной системы в соответствии с техническим заданием. ПК 5.5. Осуществлять тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы. ПК 11.1. Осуществлять сбор, обработку и анализ информации для проектирования баз данных. ПК 11.3. Разрабатывать объекты базы данных в соответствии с результатами анализа предметной области. ПК 11.4. Реализовывать базу данных в конкретной системе управления базами данных.
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	10	1	5	3	1
2.		Использование ИИ-инструментов	2	0	1	1	0
3.		Коллекции	12	1	6	4	1
4.		Функции	9	0	6	2	1
5.		Проект	2	0	2	0	0
6.		Промежуточная аттестация	1	0	0	0	1
7.	Модуль 2 Объектно-ориентированное программирование (ООП)	Объектно-ориентированное программирование (ООП)	14	1	8	4	1
8.		Парадигмы ООП	14	1	8	4	1
9.		Командный проект	7	0	4	2	1
10.		Промежуточная аттестация	1	0	0	0	1
11.	Модуль 3 WEB и API	Работа с файлами	7	1	3	2	1
12.		Ответственное использование ИИ-инструментов	2	0	1	1	0
13.		Использование библиотек	14	1	8	4	1
14.		WEB и API	6	0	4	1	1
15.		Проект	6	0	4	2	0
16.		Промежуточная аттестация	1	0	0	0	1
17.	Модуль 4 Базы данных	PyQT	16	2	9	4	1
18.		Базы данных	10	1	5	3	1
19.		Командный проект	3	0	2	1	0
20.		Кейс-задача	6	0	4	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				

Контур Крипто
владелец Марченя Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
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;	Основы Python	Знакомство с курсом, установка Python и IDE. Языки, типизация, области применения. Синтаксис, PEP 8, ввод/вывод, переменные, типы данных, преобразование типов. Арифметические операции, приоритет, форматирование строк. Логические и сравнительные операции. Условные операторы (if, elif, else, match). Циклы (for, while), range, обработка данных в цикле. Алгоритмы суммирования, подсчёта, произведения. Разложение чисел, поиск экстремумов. Управление циклом (break, continue, else, вложенные циклы).	Теоретическая подготовка	1
		Решение задач на ввод/вывод, переменные, арифметику, типы. Решение задач на условия и сложные логические выражения. Решение задач на циклы: суммирование, подсчёт, разложение чисел, поиск максимума/минимума. Решение задач с break, continue, вложенными циклами. Разработка программ, поиск и исправление ошибок.	Практическая работа	5
		Решение задач на основные конструкции Python. Анализ и отладка кода. Подготовка к текущему контролю.	Самостоятельная работа	3
		Решение задач на поиск ошибок в коде ИИ. Написание программ с использованием ИИ только для объяснения и проверки. Тесты на знание возможностей, ограничений и правил этичного использования ИИ.	Текущий контроль	1
	Использование ИИ-инструментов		Теоретическая подготовка	0
		Анализ кода, сгенерированного ИИ, поиск ошибок. Решение задач с использованием ИИ для объяснения и подсказок, но без копирования готовых решений. Разработка и проверка программ с использованием ИИ в качестве консультанта. Составление памятки по безопасной работе с ИИ.	Практическая работа	1
		Решение задач на анализ и исправление кода от ИИ. Генерация тестов с помощью ИИ. Самостоятельное написание программ с осознанным привлечением ИИ на этапах планирования и отладки.	Самостоятельная работа	1
			Текущий контроль	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

синтаксис условных конструкций (if, elif, else); синтаксис циклов for и while; 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: индексируемые, изменяемые, упорядоченные. Строки: индексы, срезы, конкатенация, умножение, сравнение, len(), ord(), chr(). Методы строк (регистр, классификация, поиск, замена). Списки: методы добавления, удаления, поиска, сортировки, функции min(), sum(), sorted(). Кортежи: неизменяемые последовательности. split() и join(). Множества: уникальность, операции (|, &, -, ^), методы (add, remove, discard). Словари: пары «ключ-значение», методы (keys, values, items, get, update, pop). Генераторы коллекций (list comprehensions, set/dict comprehensions). Преобразование коллекций (list(), tuple(), set(), dict()). Структура задачи (легенда, данные, примеры, ограничения). Сложность алгоритмов: O(1), O(n), O(n²). Алгоритмы: Евклида (НОД), перевод в двоичную систему.	Теоретическая подготовка	1
		Решение задач на строки (индексы, срезы, методы). Решение задач на списки (добавление, удаление, сортировка, поиск). Решение задач на множества (удаление дубликатов, операции). Решение задач на словари (подсчёт частоты, группировка). Решение задач на генераторы коллекций и преобразование типов. Решение задач на анализ структуры, сложности алгоритмов, выбор эффективного решения. Решение задач с алгоритмом Евклида и переводом в двоичную систему.	Практическая работа	6
		Решение задач на обработку строк, списков, множеств, словарей. Решение задач на split(), join(), генераторы коллекций. Решение задач на определение сложности алгоритмов. Решение задач с применением алгоритма Евклида. Анализ и отладка кода.	Самостоятельная работа	4
		Решение задач на коллекции (строки, списки, множества, словари). Выполнение тестов на методы коллекций и сложность алгоритмов. Решение комплексных задач, комбинирующих типы коллекций.	Текущий контроль	1
	Функции		Теоретическая подготовка	0
		Решение задач на создание функций с параметрами и возвратом значений. Решение задач на *args/**kwargs и лямбда-функции. Решение задач на map(), filter(), reduce(). Решение задач на функции-генераторы и генераторные выражения. Решение задач на рекурсию (факториал, числа Фибоначчи). Решение задач на области видимости (global, nonlocal). Решение задач на замыкания и декораторы. Решение задач на импорт и использование модулей (math, calendar, itertools). Решение задач на создание собственных модулей и пакетов, установку сторонних пакетов (pip).	Практическая работа	6
		Решение задач на функции и лямбда-выражения. Решение задач на функции высшего порядка и генераторы. Решение задач на рекурсивные алгоритмы. Решение задач на замыкания, декораторы и области видимости. Решение задач на импорт и использование модулей, создание пакетов. Анализ и отладка кода с функциями.	Самостоятельная работа	2
		Решение задач на вычисление площади круга и длины окружности с использованием math.pi. Решение задач на рекурсивное вычисление суммы цифр числа. Решение задач на генерацию сочетаний с помощью itertools.combinations. Решение задач на определение дня недели с помощью calendar.weekday. Решение задач на фильтрацию чисел, кратных указанному делителю (функция с *args и именованным аргументом).	Текущий контроль	1
	Проект		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

		Реализация ввода и проверки города пользователем. Реализация хода компьютера (поиск, вывод, обновление буквы). Реализация проверки завершения игры. Тестирование и отладка программы.	Практическая работа	2
			Самостоятельная работа	0
			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. 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();	Объектно-ориентированное программирование (ООП)	Парадигмы программирования, мультипарадигменность Python. ООП: класс, объект, атрибуты, методы. Функции type(), isinstance(), класс object. Создание классов (class), именование (CamelCase). Атрибуты класса и экземпляра, __dict__, поиск атрибутов. setattr(), getattr(), delattr(), hasattr(). __doc__. Методы: self, @classmethod, @staticmethod. Специальные методы: __init__(), __del__(), __str__(), __repr__(). Перегрузка операторов: арифметические (__add__, __sub__, __mul__, __truediv__ и др.), расширенные (__iadd__), отражённые (__radd__). Сравнения: __eq__, __ne__, __lt__, __le__, __gt__, __ge__, @total_ordering. Другие: __getitem__, __setitem__, __delitem__, __len__, __contains__, __call__, __abs__, __bool__. Практика: проектирование классов для игр (на примере «Крестики-нолики»).	Теоретическая подготовка	1
		Решение задач на создание классов, атрибутов, методов (type, isinstance, setattr, getattr, delattr, hasattr). Решение задач на @classmethod, @staticmethod, self. Решение задач на __init__, __str__, __repr__. Решение задач на перегрузку арифметических операций и сравнений (Fraction, Vector). Решение задач на __getitem__, __setitem__, __len__. Решение задач на разработку игры «Крестики-нолики» с использованием ООП. Анализ и отладка кода.	Практическая работа	8
		Решение задач на создание классов и объектов. Решение задач на специальные методы и перегрузку операторов. Решение задач на разработку игр с применением ООП. Изучение документации по специальным методам. Подготовка к текущему контролю.	Самостоятельная работа	4
		Тесты на синтаксис классов, методов, декораторов. Решение задач на определение результатов программ с классами. Исправление ошибок в коде с классами и перегрузкой операторов. Комплексные задачи на иерархии классов.	Текущий контроль	1
	Парадигмы ООП	Наследование: базовый и дочерний классы, синтаксис, переопределение методов, super(), наследование от встроенных типов. Множественное наследование, MRO (__mro__), issubclass(), isinstance(). Полиморфизм: одноименные методы, утиная типизация, переопределение. Инкапсуляция: публичные, защищённые (_), приватные (__, name mangling). Геттеры, сеттеры, @property, __slots__. Специальные методы для доступа к атрибутам: __setattr__, __getattribute__, __getattr__, __delattr__. GUI: Tkinter (Tk, Label, Button, pack(), mainloop()). Исключения: try...except..., else, finally, raise. Собственные исключения (наследование от Exception). Стратегии LBYL и EAFP.	Теоретическая подготовка	1

Контур Крипто
владелец Мартынов Виталий Вадимович
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3191c94f2cb04c6f27cc019aeaad
срок действия 25.12.2025 - 05.03.2027

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 наследование, переопределение методов, super(). Решение задач на множественное наследование и MRO. Решение задач на полиморфизм (единый интерфейс для разных классов). Решение задач на инкапсуляцию: геттеры/сеттеры, @property, __slots__. Решение задач на создание GUI-приложений на Tkinter. Решение задач на обработку исключений и создание собственных исключений. Разработка игры «Крестики-нолики» с ООП и интеграцией Tkinter. Анализ и отладка кода.	Практическая работа	8
		Решение задач на наследование и полиморфизм (геометрические фигуры). Решение задач на инкапсуляцию. Решение задач на обработку исключений. Решение задач на создание GUI-приложений. Выполнение заданий на переопределение методов, super(), @property. Мини-проект с применением принципов ООП. Изучение документации по Tkinter и исключениям.	Самостоятельная работа	4
		Решение задач на MRO, наследование, полиморфизм. Решение задач на обработку исключений. Тесты по синтаксису ООП и Tkinter. Исправление ошибок в коде. Комплексные задачи на иерархии классов.	Текущий контроль	1
	Командный проект		Теоретическая подготовка	0
		Задачи на иерархии классов, super(), переопределение методов. Задачи на множественное наследование и MRO. Задачи на полиморфизм (единый интерфейс). Задачи на инкапсуляцию: @property, __slots__. Задачи на Tkinter (окна, кнопки, события). Задачи на обработку и создание исключений. Разработка игры «Крестики-нолики» с ООП и Tkinter.	Практическая работа	4
		Задачи на наследование и полиморфизм (фигуры, системы счисления). Задачи на инкапсуляцию (управление атрибутами). Задачи на исключения (калькулятор, проверка пароля). Задачи на Tkinter (простые приложения). Мини-проект с применением ООП.	Самостоятельная работа	2
		Задачи на MRO, наследование, полиморфизм, исключения. Тесты по синтаксису ООП, Tkinter. Исправление ошибок в коде. Комплексные задачи на иерархии классов.	Текущий контроль	1
	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 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 научатся 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), логирование (logging), основы работы с API	Работа с файлами	Стандартный поток ввода (sys.stdin), чтение данных (read(), readlines()). Функция open(), режимы (r, w, a), менеджер контекста with. Чтение (readlines()) и запись (write(), writelines()). Формат JSON: модуль json (load, loads, dump, dumps), параметры (ensure_ascii, indent, sort_keys). Формат CSV: модуль csv (reader, DictReader, writer, DictWriter), параметры (delimiter, quotechar). Преобразование между JSON и CSV.	Теоретическая подготовка	1
		Задачи на чтение/запись текстовых файлов. Задачи на чтение/парсинг JSON и CSV. Задачи на запись данных в JSON и CSV. Задачи на преобразование между форматами.	Практическая работа	3

Контур.Крипто владелец Мартышева Виктория Вадимовна
Документ подписан квалифицированной электронной подписью до 04.08.2026
серийный номер 01ba3191c94f2eb04c6f27cc019aedad
срок действия 06.12.2025 - 06.03.2027

(requests), парсинг данных (HTML-разбор), основы работы с библиотекой vk_api, систему контроля версий Git, а также выполняют проект «VK бот» и знакомятся с принципами ответственного использования технологий искусственного интеллекта. 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. Планируемые результаты обучения 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. принципы ответственного использования ИИ-инструментов. 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; использовать ИИ-инструменты для анализа и оптимизации кода; выполнять командный проект; владеть: навыками работы с файлами и внешними данными; навыками обработки данных в форматах JSON и CSV;		Чтение и обработка текстовых файлов. Чтение JSON и CSV, извлечение информации. Запись данных в JSON и CSV. Преобразование форматов.	Самостоятельная работа	2
		Тесты на знание ограничений, этики, безопасности ИИ. Задачи на анализ и исправление кода, написание программ с осознанным использованием ИИ. Комплексные задачи с привлечением ИИ в качестве консультанта.	Текущий контроль	1
	Ответственное использование ИИ-инструментов		Теоретическая подготовка	0
		Анализ и исправление кода, сгенерированного ИИ. Решение задач с использованием ИИ для объяснений и подсказок, самостоятельная реализация. Генерация тестов с помощью ИИ, выявление ошибок. Разработка программ с осознанным использованием ИИ на этапах планирования и отладки (без копирования готового кода).	Практическая работа	1
		Задачи на анализ и исправление кода ИИ. Задачи на применение ИИ для генерации тестов. Написание программ с использованием ИИ для подсказок, но с самостоятельной реализацией. Составление памятки по безопасной работе с ИИ.	Самостоятельная работа	1
			Текущий контроль	0
	Использование библиотек	Стандартный поток ввода (sys.stdin), чтение данных (read(), readlines()). Функция open(), режимы, менеджер контекста with. Чтение и запись файлов. Форматы JSON и CSV: модули json и csv, методы чтения/записи, преобразование между форматами. Командная строка (CLI): навигация, создание/удаление файлов, запуск скриптов, sys.argv. Модуль argparse: позиционные и опциональные аргументы, add_argument(). Модуль datetime: классы date, time, datetime, timedelta, форматирование (strftime, strptime). Модуль schedule: периодические задачи. Регулярные выражения: метасимволы, квантификаторы, группы. Модуль re: search, match, findall, sub, split. Тестирование: assert, doctest, unittest (TestCase, методы проверки). Логирование: модуль logging, уровни, настройка (basicConfig), обработчики (StreamHandler, FileHandler), форматтеры, логирование исключений. Ответственное использование ИИ: этика, проверка кода, безопасность.	Теоретическая подготовка	1
		Задачи на чтение/запись файлов, JSON и CSV. Задачи на работу с командной строкой и argparse. Задачи на работу с датой и временем (datetime, schedule). Задачи на использование регулярных выражений (re). Задачи на написание тестов (assert, doctest, unittest). Задачи на настройку логирования (logging). Задачи на анализ и исправление кода с использованием тестирования.	Практическая работа	8
		Решение задач на чтение/запись файлов, работу с JSON и CSV. Решение задач на argparse, datetime, schedule, re. Решение задач на написание тестов и настройку логирования. Решение задач с осознанным использованием ИИ для поиска ошибок и генерации тестов.	Самостоятельная работа	4
		Выполнение заданий текущего контроля: автоматизированная проверка кода.	Текущий контроль	1
		Задания на argparse (калькулятор, генератор последовательности). Задания на datetime (расчёт возраста). Задания на re (поиск email-адресов). Задания на поиск и исправление ошибок с использованием ИИ.		
	WEB и API		Теоретическая подготовка	0

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

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 применения ИИ-инструментов в процессе разработки решений. навыками командной проектной деятельности.		Задачи на файлы, JSON, CSV. Задачи на CLI и argparse. Задачи на datetime и schedule. Задачи на регулярные выражения (re). Задачи на тестирование (assert, doctest, unittest). Задачи на логирование (logging). Задачи на выполнение запросов к API (requests). Задачи на парсинг веб-страниц (Beautiful Soup).	Практическая работа	4
		Решение задач на файлы, JSON, CSV. Решение задач на argparse, datetime, schedule, re. Решение задач на тестирование и логирование. Решение задач на работу с API (Работа России, DaData). Решение задач на парсинг страниц (PEP, PyPI, Python docs).	Самостоятельная работа	1
		Задачи на работу с Git: создание репозитория, коммиты, push, pull, README.md. Задачи на анализ и исправление ошибок в коде бота. Комплексные задачи на интеграцию Git, VK API и внешних API.	Текущий контроль	1
	Проект		Теоретическая подготовка	0
		Задачи на создание репозитория, клонирование, коммиты, push, README.md. Задачи на создание VK-бота (эхо-бот, бот с проверкой орфографии через Яндекс Спеллер). Задачи на настройку виртуального окружения, установку зависимостей, requirements.txt. Задачи на работу с Git через OpenIDE (клонирование, коммиты, push, pull). Разработка и отладка кода бота, тестирование сценариев (корректный/некорректный ввод).	Практическая работа	4
		Настройка репозитория на GitVerse, клонирование, коммиты, push, работа с README.md. Создание VK-бота (эхо-бот) с использованием vk_api. Создание VK-бота с интеграцией Яндекс Спеллера (проверка орфографии, отправка результатов). Настройка виртуального окружения, установка зависимостей, создание requirements.txt. Работа с Git через OpenIDE (клонирование, коммиты, push, pull). Отладка кода бота, логирование событий с использованием logging.	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 Маршеня Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Модуль 4 Базы данных Модуль направлен на формирование у обучающихся практических навыков разработки графических интерфейсов и работы с базами данных в языке Python, освоение принципов построения пользовательских приложений, обработки событий и взаимодействия с хранилищами данных. В рамках модуля обучающиеся изучают создание интерфейсов с использованием библиотеки PyQt (виджеты: QWidget, QLabel, QPushButton, QLineEdit, QTableWidget и другие), работу с Qt Designer, обработку событий (signals/slots), создание окон и диалогов, основы баз данных и языка SQL, выполнение SQL-запросов (SELECT, INSERT, UPDATE, DELETE и другие), работу с базами данных в Python (sqlite3), а также выполняют командный проект «Список задач» и решают кейс-задачу «Подготовка бейджей для мероприятий». Задачи модуля сформировать навыки разработки графических интерфейсов с использованием библиотеки PyQt; познакомить с основными виджетами (QWidget, QLabel, QPushButton, QLineEdit, QTableWidget и другие); сформировать навыки использования Qt Designer для проектирования интерфейсов; познакомить с обработкой событий (signals/slots); сформировать навыки создания окон и диалогов в приложениях; сформировать навыки работы с базами данных; познакомить с основами SQL (SELECT, INSERT, UPDATE, DELETE и другие); сформировать навыки выполнения SQL-запросов; познакомить с работой с базами данных в Python (sqlite3); развить навыки разработки прикладных приложений с графическим интерфейсом; развить навыки проектной деятельности при выполнении командного проекта; развить навыки решения прикладной кейс-задачи. Планируемые результаты обучения По итогам освоения модуля обучающийся будет: знать: основы разработки графических интерфейсов с использованием PyQt; назначение основных виджетов (QWidget, QLabel, QPushButton, QLineEdit, QTableWidget и другие); принципы работы Qt Designer; основы обработки событий (signals/slots); основы работы с окнами и диалогами; основы баз данных; основы языка SQL (SELECT, INSERT, UPDATE, DELETE и другие); основы работы с sqlite3 в Python. уметь: создавать графические интерфейсы с использованием PyQt; использовать основные виджеты (QLabel, QPushButton, QLineEdit, QTableWidget); применять Qt Designer для проектирования интерфейсов; обрабатывать события через signals/slots;	PyQT	GUI: понятие, история. Библиотеки Python: Tkinter (стандартная), PyQt (установка, модули QtWidgets/QtCore/QtGui). Виджеты PyQt: базовые (QLabel, QPushButton, QLineEdit, QSpinBox, QComboBox, QTextEdit и др.). Позиционирование: абсолютное (resize, move) и менеджеры макетов (QHBoxLayout, QVBoxLayout, QGridLayout, QFormLayout), растяжители (addStretch, addSpacing). Сигналы и слоты: соединение (connect), основные сигналы (clicked, textChanged, currentIndexChanged). События: обработчики (keyPressEvent, mousePressEvent). Главное окно: QMainWindow (меню, панели инструментов, строка состояния, док-виджеты). Меню (QMenuBar, QAction), панели (QToolBar, QIcon), статус (QStatusBar). Диалоги: QDialog (модальные/немодальные), QMessageBox, QInputDialog, QFileDialog. Qt Designer: .ui файлы, подключение через uic.loadUi() или pyuic6. Практика: калькулятор, текстовый редактор, конвертер, генератор паролей.	Теоретическая подготовка	2
		Задачи на создание оконных приложений (Tkinter, PyQt). Задачи на размещение виджетов (макеты и абсолютное позиционирование). Задачи на работу с виджетами (поля ввода, списки, счётчики). Задачи на обработку сигналов/событий (нажатия кнопок, ввод текста, клавиатура/мышь). Задачи на создание главного окна с меню, панелями, статусом. Задачи на создание диалоговых окон (сообщения, ввод данных, выбор файлов). Задачи на использование Qt Designer и подключение интерфейсов.	Практическая работа	9
		Создание приложений на PyQt: калькулятор, текстовый редактор, конвертер валют, перевод чисел, генератор паролей. Использование Qt Designer для интерфейсов (анализ текста, конвертер). Обработка сигналов, слотов, событий. Создание меню, панелей инструментов, строки состояния, диалогов. Изучение документации по PyQt6.	Самостоятельная работа	4
		Задачи на создание приложений (номеронабиратель, выбор месяца, конвертер °C→°F, генератор пароля, форма авторизации). Выполнение заданий текущего контроля: автоматическая проверка кода на соответствие требованиям (интерфейс, корректность работы).	Текущий контроль	1
	Базы данных	БД: реляционные/нереляционные, СУБД (SQLite). SQL: SELECT, WHERE, ORDER BY, GROUP BY, HAVING, JOIN, подзапросы. Типы связей (1:1, 1:M, M:M), первичный/внешний ключи, ограничения. Агрегатные функции (COUNT, SUM, AVG, MIN, MAX). Модификация данных: INSERT, UPDATE, DELETE, каскадное удаление. Модуль sqlite3: подключение (connect), выполнение запросов (execute, executemany), извлечение данных (fetchone, fetchall), транзакции (commit, with), исключения. Интеграция с PyQt: вывод таблиц (QTableWidget), списки (QListWidget), выбор (QComboBox). CRUD-операции с подтверждением через QMessageBox. Многотабличные запросы, lastrowid.	Теоретическая подготовка	1
		Задачи на SQL-запросы (выборка, фильтрация, сортировка, группировка, JOIN). Задачи на модификацию данных (INSERT, UPDATE, DELETE). Задачи на создание таблиц с ограничениями. Задачи на подключение и работу с БД через sqlite3. Задачи на создание PyQt-приложений для работы с БД (вывод, добавление, редактирование, удаление).	Практическая работа	5
		Написание SQL-запросов для работы с таблицами. Создание Python-скриптов с sqlite3. Разработка PyQt-приложений для работы с БД (проект «База данных фильмов»: методы showMovies, addMovie, editMovie, deleteMovie). Изучение документации по SQLite3 и sqlite3.	Самостоятельная работа	3

Контур Крипто
владелец Маркелов Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3197c941c2cb04c6d127cc0fa2e6ad
срок действия 25.10.2025 - 25.10.2026

создавать окна и диалоги; выполнять SQL-запросы (SELECT, INSERT, UPDATE, DELETE и другие); работать с базами данных в Python (sqlite3); разрабатывать прикладные приложения с графическим интерфейсом; решать прикладную кейс-задачу. владеть: навыками разработки GUI-приложений на PyQt; навыками работы с виджетами и интерфейсами; навыками проектирования интерфейсов в Qt Designer; навыками обработки событий; навыками работы с базами данных; навыками выполнения SQL-запросов; навыками разработки прикладных программных решений; навыками командной проектной деятельности.).		Решение задач на написание SQL-запросов с SELECT, WHERE, ORDER BY. Решение задач на соединение таблиц (JOIN) и агрегацию. Решение задач на модификацию данных (INSERT, UPDATE, DELETE). Автоматическая проверка SQL-запросов.	Текущий контроль	1
	Командный проект		Теоретическая подготовка	0
		Задачи на реализацию методов классов Categories и Tasks (загрузка, добавление, удаление, редактирование, фильтрация). Задачи на проектирование БД, CRUD-операции, обработку сигналов, диалоговые окна. Задачи на анализ кейс-задачи, формулирование проблемы, целей, результатов. Задачи на выбор технологий (библиотеки для PDF, QR-кодов, изображений). Разработка прототипа/MVP решения кейс-задачи.	Практическая работа	2
		Разработка проекта «Список задач» (реализация всех методов, тестирование). Разработка решения кейс-задачи «Подготовка бейджей» (архитектура, реализация, тестирование). Оформление кода, создание README.md. Подготовка к защите: план выступления, демонстрация, ответы на вопросы. Изучение документации по библиотекам для PDF, изображений, QR-кодов.	Самостоятельная работа	1
			Текущий контроль	0
	Кейс-задача		Теоретическая подготовка	0
		Задачи на методы классов Categories и Tasks (CRUD, фильтрация). Задачи на проектирование БД, обработку сигналов, диалоговые окна. Задачи на анализ кейс-задачи, постановку целей, выбор технологий. Разработка прототипа/MVP кейс-задачи.	Практическая работа	4
		Разработка проекта «Список задач» (все методы, тестирование). Разработка решения кейс-задачи (архитектура, реализация, тестирование). Оформление кода, README.md, подготовка к защите. Изучение библиотек для PDF, изображений, QR-кодов.	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
				Объем в ак.ч. / Объем в %

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Марцинив Виктория Вадимовна
серийный номер 01ab1979e7c1c8b0a82e8a19aeba
срок действия 05.12.2025 - 05.03.2027

ИТОГО ПО ПРОГРАММЕ:	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, СПО)	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
владелец Маршеня Виктория Вадимовна
Документ подписан усиленной квалифицированной электронной подписью 04.06.2026
серийный номер 01ac2191a94f9ab04c6f2b6c810ea4ad
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 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.	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. SQLiteStudio версии 3.4.0 и выше. Qt Designer. 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://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 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	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 дркументация по библиотекам Python https://proproprogs.ru/python — ресурс с информацией о программировании на Python https://doc.qt.io/ — официальная документация по Qt https://www.riverbankcomputing.com/static/Docs/PyQt6/ — официальная документация по PyQt6

Наименование поля	Значение полей	Значение полей	Значение полей	Значение полей
Порядковый номер модуля	1	2	3	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

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 https://leetcode.com/ — задачи по программированию	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 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://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	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

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 Маршова Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
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
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
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

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

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)

Варианты заданий	Задача 1. Сумма двух чисел с проверкой на чётность
	Напиши программу, которая запрашивает два целых числа и выводит их сумму. Если сумма чётная, программа должна вывести строку «Сумма чётная», иначе – «Сумма нечётная».
	Входные данные
	Два целых числа, каждое на отдельной строке.
	Выходные данные
	Сначала сумма (целое число), затем на новой строке сообщение о чётности.
	Пример
	Ввод:
	3
	5
	Вывод:
	8
	Сумма чётная
	Примечание
	Подсказки внутри input() не используй.
	Решение
	a = int(input())
	b = int(input())
	s = a + b
	print(s)
	if s % 2 == 0:
	  print('Сумма чётная')
	else:
	  print('Сумма нечётная')
	Задача 2. Проверка числа на кратность 5 и чётность
	Напиши программу, которая запрашивает целое число и выводит:
	«Кратно 5 и чётно», если число делится на 5 и на 2;
	«Кратно 5, но не чётно», если число делится на 5, но не на 2;
	«Не кратно 5» в остальных случаях.
	Входные данные
	Одно целое число.
	Выходные данные
	Одна строка.
	Пример
	Ввод:
	10
	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

Кратно 5 и чётно
Примечание
Подсказки внутри input() не используй.
Решение

```
n = int(input())
if n % 5 == 0 and n % 2 == 0:
    print('Кратно 5 и чётно')
elif n % 5 == 0:
    print('Кратно 5, но не чётно')
else:
    print('Не кратно 5')
```

Задача 3. Наименьшее из трёх чисел
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 4. Количество чётных чисел до нуля

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 целые числа (каждое на новой строке) до тех пор, пока не будет введён 0. Программа должна подсчитать, сколько среди введённых чисел было чётных (не считая завершающего нуля), и вывести это количество.
	Входные данные
	Несколько целых чисел (каждое на новой строке). Последнее число – 0.
	Выходные данные
	Одно целое число – количество чётных чисел.
	Пример
	Ввод:
	2
	5
	4
	8
	0
	Вывод:
	3
	Примечание
	Подсказки внутри input() не используй.
	Решение
	count = 0
	while True:
	  x = int(input())
	  if x == 0:
	  break
	  if x % 2 == 0:
	  count += 1
	print(count)
	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

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

Формы контроля	Практические задания на исправление ошибок в готовом программном коде Практические задания на написание программного кода по предложенному алгоритму и описанию задачи
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. Создание аббревиатуры из строки
	Напиши программу, которая запрашивает у пользователя строку, содержащую несколько слов, разделённых пробелами. Программа должна сформировать аббревиатуру из первых букв каждого слова, переведённых в верхний регистр, и вывести полученную строку. Если пользователь ввёл пустую строку, программа должна вывести пустую строку (без ошибок). Гарантируется, что слова разделены одним пробелом.
	Входные данные:
	Одна строка – слова через пробел.
	Выходные данные:
	Одна строка – аббревиатура (заглавными буквами).
	Пример ввода:
	программирование на python
	Пример вывода:
	ПНП
	Решение:
	s = input()
	if s:
	abbr = ''
	for word in s.split():
	abbr += word[0].upper()
	print(abbr)
	2. Разделение чисел на чётные и нечётные (списки)
	Напиши программу, которая запрашивает 8 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:
	8 целых чисел (по одному на строке).
	Выходные данные:
	Две строки: список чётных и список нечётных чисел.
	Пример ввода:
	5
	2
	8
	3
	1
	6
	4
	7
	Пример вывода:
	[2, 8, 6, 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

[5, 3, 1, 7]
Решение:

```
evens = []
odds = []
for _ in range(8):
    x = int(input())
    if x % 2 == 0:
        evens.append(x)
    else:
        odds.append(x)
print(evens)
print(odds)
```

3. Количество уникальных символов в строке
Напиши программу, которая запрашивает строку и выводит количество различных символов (уникальных) в этой строке. Пробелы и знаки препинания считаются символами.
Входные данные:
Одна строка.
Выходные данные:
Одно целое число – количество уникальных символов.
Пример ввода:
Привет, мир!
Пример вывода:
11
(Уникальные символы: П, р, и, в, е, т, ,, , м, ! = 11)
Решение:

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

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 ввода:
яблоко груша банан груша апельсин банан груша
Пример вывода:
груша
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
```

5. Сумма чисел от 1 до n (арифметическая прогрессия)
Напиши программу, которая запрашивает целое положительное число n и вычисляет сумму всех целых чисел от 1 до n включительно. Используй формулу арифметической прогрессии, чтобы программа работала быстро даже для больших n (без циклов).
Входные данные:
Одно целое число n ($1 \le n \le 10^9$).
Выходные данные:
Одно целое число – сумма.
Пример ввода:
100
Пример вывода:
5050
Решение:

```
n = int(input())
total = (1 + n) * n // 2
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

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. Площадь круга и длина окружности (модуль math) Напиши программу, которая запрашивает радиус круга (вещественное число) и с помощью константы math.pi вычисляет: площадь круга ($S = \pi * R^2$); длину окружности ($C = 2 * \pi * R$). Результаты выведи на отдельных строках без округления. Входные данные Одно вещественное число – радиус. Выходные данные Два вещественных числа, каждое на новой строке. Пример Ввод: 4 Вывод: 50.26548245743669 25.132741228718345 Примечание Подсказки внутри input() не используй. Решение import math r = float(input()) area = math.pi * r ** 2 circumference = 2 * math.pi * r print(area) print(circumference) Задача 2. Рекурсивная сумма цифр числа Напиши рекурсивную функцию sum_of_digits(n), которая принимает целое положительное число и возвращает сумму его цифр. Используй рекурсию (без циклов). В основной программе запроси число и выведи результат. Входные данные Одно целое положительное число. Выходные данные Одно целое число – сумма цифр. Пример Ввод: 2345 Вывод: 14

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

Примечание
Подсказки внутри input() не используй.
Решение

```
def sum_of_digits(n):
    if n == 0:
        return 0
    return n % 10 + sum_of_digits(n // 10)

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

Задача 3. Генерация сочетаний (модуль itertools)
Напиши программу, которая запрашивает целое число n (количество учеников) и выводит все возможные пары учеников для дежурства (сочетания по 2 из n).
Пары должны выводиться в порядке возрастания номеров. Используй itertools.combinations.
Если n меньше 2, программа должна вывести пустую строку (без ошибок).
Входные данные
Одно целое число n ($1 \le n \le 20$).
Выходные данные
Пары номеров через пробел (каждая пара на новой строке), отсортированные по возрастанию.
Пример
Ввод:
4
Вывод:
1 2
1 3
1 4
2 3
2 4
3 4
Примечание
Подсказки внутри input() не используй.
Решение

```
import itertools

n = int(input())
if n >= 2:
    for pair in itertools.combinations(range(1, n + 1), 2):
        print(pair[0], pair[1])
```

Задача 4. Определение дня недели (модуль calendar)

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 дату (день, месяц, год – три целых числа через пробел) и выводит название дня недели, на который приходится эта дата. Используй модуль calendar. Список дней недели на русском языке: Понедельник, Вторник, Среда, Четверг, Пятница, Суббота, Воскресенье. Входные данные Три целых числа через пробел – день, месяц, год. Выходные данные Одна строка – название дня недели. Пример Ввод: 1 1 2024 Вывод: Понедельник Примечание Подсказки внутри input() не используй. Решение import calendar weekdays = ['Понедельник', 'Вторник', 'Среда', 'Четверг', 'Пятница', 'Суббота', 'Воскресенье'] day, month, year = map(int, input().split()) weekday_index = calendar.weekday(year, month, day) # 0 = понедельник print(weekdays[weekday_index]) Задача 5. Фильтрация чисел, кратных указанному делителю (функция) Напиши функцию filter_divisible(*args, divisor), которая принимает произвольное количество целых чисел (позиционные аргументы) и один именованный аргумент divisor – целое число, большее нуля. Функция должна возвращать список тех чисел, которые делятся на divisor без остатка. В основной программе прочитай строку целых чисел через пробел, затем целое число – делитель. Вызови функцию и выведи результат (список в формате Python). Входные данные Первая строка – целые числа через пробел. Вторая строка – целое число divisor. Выходные данные Список чисел, кратных divisor. Пример Ввод: 10 15 20 25 30 5 Вывод: [10, 15, 20, 25, 30] Примечание Подсказки внутри input() не используй.
	Решение def filter_divisible(*args, divisor): return [x for x in args if x % divisor == 0] numbers = list(map(int, input().split())) d = int(input()) print(filter_divisible(*numbers, divisor=d))
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

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Маршова Виктория Вадимовна
серийный номер 1f6ab19f1c94f2cb04c6f27cc019aedad
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. Класс «Книга» с информацией о прочтении
	Создай класс Book с атрибутами: title (название), author (автор), pages (количество страниц). Реализуй:
	конструктор __init__(title, author, pages);
	метод info(), который выводит строку в формате:
	«{автор} — {название} ({страницы} стр.)».
	В основной программе создай экземпляр класса Book с данными, введёнными с клавиатуры (три строки: название, автор, количество страниц). Выведи результат вызова метода info().
	Входные данные:
	Три строки: название, автор, количество страниц (целое число).
	Выходные данные:
	Одна строка.
	Пример ввода:
	Война и мир
	Толстой
	1425
	Пример вывода:
	Толстой — Война и мир (1425 стр.)
	Решение:
	class Book:
	  def __init__(self, title, author, pages):
	  self.title = title
	  self.author = author
	  self.pages = pages
	
	  def info(self):
	  print(f'{self.author} — {self.title} ({self.pages} стр.)')
	
	title = input()
	author = input()
	pages = int(input())
	book = Book(title, author, pages)
	book.info()
	2. 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(), который 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 имя и курс через пробел. Создай объект, примени метод promote() и выведи строковое представление студента.

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:
Одна строка: имя и курс (целое число) через пробел.
Выходные данные:
Сообщение об исправлении курса (если нужно) и строковое представление студента после promote().
Пример ввода:
Анна 5
Пример вывода:
Курс исправлен на 1
Анна, курс 2
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}'

name, course = input().split()
s = Student(name, int(course))
s.promote()
print(s)
```

3. Класс «Прямоугольник» с инкапсуляцией
Создай класс Rectangle с приватными атрибутами __width и __height. Реализуй:
конструктор __init__(width, height), который принимает положительные числа; если переданное значение меньше или равно нулю, выводи «Размер должен быть больше 0» и установи значение 1;
геттеры width и height через @property, возвращающие значения;
сеттеры width и height через @setter, которые проверяют, что новое значение положительное; если нет – выводи «Размер должен быть больше 0» и не меняй значение;

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

метод area(), возвращающий площадь прямоугольника;
метод __str__, возвращающий строку «Прямоугольник {ширина}×{высота}».
В основной программе прочитай два целых числа (ширину и высоту) через пробел. Создай объект, выведи его площадь и строковое представление.
Входные данные:
Два целых числа (ширина и высота) через пробел.
Выходные данные:
Площадь и строковое представление прямоугольника.
Пример ввода:
10 5
Пример вывода:
Площадь: 50
Прямоугольник 10×5
Решение:

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

    @property
    def width(self):
        return self.__width

    @width.setter
    def width(self, value):
        if value <= 0:
            print('Размер должен быть больше 0')
            self.__width = 1
        else:
            self.__width = value

    @property
    def height(self):
        return self.__height

    @height.setter
    def height(self, value):
        if value <= 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('Размер должен быть больше 0')
            self.__height = 1
        else:
            self.__height = value

    def area(self):
        return self.__width * self.__height

    def __str__(self):
        return f'Прямоугольник {self.__width}×{self.__height}'

w, h = map(int, input().split())
rect = Rectangle(w, h)
print(f'Площадь: {rect.area()}')
print(rect)
```

4. 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
```

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

```
        self.price = price

    @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))
```

5. Класс «Время» с операцией сложения

Создай класс Time для представления времени в часах и минутах. Конструктор принимает часы и минуты (целые числа). Реализуй:
метод __str__, возвращающий строку в формате «{часы}:{минуты}» (минуты всегда двузначные, например 3:05);
метод __add__, который складывает два объекта Time и возвращает новый объект Time (сумма часов и минут, при переполнении минут они переходят в часы);
метод to_minutes(), возвращающий общее количество минут.
В основной программе прочитай две строки, каждая содержит часы и минуты через пробел. Создай два объекта Time, сложи их и выведи результат.
Входные данные:
Две строки: каждая содержит два целых числа (часы и минуты).
Выходные данные:
Одна строка – результат сложения в формате «часы:минуты».
Пример ввода:
2 30
3 45
Пример вывода:
6:15
Решение:

```
class Time:
    def __init__(self, hours, minutes):
        self.hours = hours
        self.minutes = minutes

    def to_minutes(self):
        return self.hours * 60 + self.minutes

    def __add__(self, other):
        total = self.to_minutes() + other.to_minutes()
        h = total // 60
        m = total % 60
        return Time(h, m)

    def __str__(self):
        return f'{self.hours}:{self.minutes:02d}'

h1, m1 = map(int, input().split())
t1 = Time(h1, m1)
h2, m2 = map(int, input().split())
t2 = Time(h2, m2)
print(t1 + t2)
```

Контур Крипто
владелец Маршеня Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba2191e94f2eb04e6f27ee019aedad
срок действия 05.12.2025 - 05.03.2027

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

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: классы Person и Employee
	Создай базовый класс Person с методом __init__(name, age), сохраняющим атрибуты name и age, и методом get_info(), возвращающим строку «Человек: {name}, возраст: {age}». Создай класс Employee (сотрудник), наследующий от Person. В Employee добавь атрибут position (должность) и переопредели метод get_info(), чтобы он возвращал «Сотрудник: {name}, возраст: {age}, должность: {position}». Используй super() для вызова родительского конструктора.
	Считай с клавиатуры три строки: имя, возраст (целое число), должность. Создай экземпляр Employee и выведи результат вызова get_info().
	Входные данные:
	Три строки: имя, возраст, должность.
	Выходные данные:
	Одна строка – информация о сотруднике.
	Пример ввода:
	Иванов Иван
	30
	Менеджер
	Пример вывода:
	Сотрудник: Иванов Иван, возраст: 30, должность: Менеджер
	Решение:
	class Person:
	def __init__(self, name, age):
	self.name = name
	self.age = age
	
	def get_info(self):
	return f'Человек: {self.name}, возраст: {self.age}'
	
	class Employee(Person):
	def __init__(self, name, age, position):
	super().__init__(name, age)
	self.position = position
	
	def get_info(self):
	return f'Сотрудник: {self.name}, возраст: {self.age}, должность: {self.position}'
	
	name = input()
	age = int(input())
	position = input()
	emp = Employee(name, age, position)
	print(emp.get_info())
	2. Множественное наследование: классы с действиями
	Создай классы:

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

Flyer с методом fly(), возвращающим «Летит».
Swimmer с методом swim(), возвращающим «Плывёт».
Walker с методом walk(), возвращающим «Идёт».
Создай классы, наследующие от соответствующих классов:
Duck (утка) – наследует от Flyer, Swimmer, Walker.
Penguin (пингвин) – наследует от Swimmer, Walker.
Eagle (орёл) – наследует от Flyer, Walker.
Считай с клавиатуры название класса (Duck, Penguin или Eagle), создай объект и выведи все доступные действия (вызовы методов) в порядке: сначала методы родительских классов в порядке их перечисления в определении класса, каждое действие на отдельной строке.
Входные данные:
Одна строка – название класса.
Выходные данные:
Строки с результатами методов.
Пример ввода:
Duck
Пример вывода:
Летит
Плывёт
Идёт
Решение:

```
class Flyer:
    def fly(self):
        return 'Летит'

class Swimmer:
    def swim(self):
        return 'Плывёт'

class Walker:
    def walk(self):
        return 'Идёт'

class Duck(Flyer, Swimmer, Walker):
    pass

class Penguin(Swimmer, Walker):
    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 Eagle(Flyer, Walker):
    pass

name = input()
if name == 'Duck':
    obj = Duck()
    print(obj.fly())
    print(obj.swim())
    print(obj.walk())
elif name == 'Penguin':
    obj = Penguin()
    print(obj.swim())
    print(obj.walk())
elif name == 'Eagle':
    obj = Eagle()
    print(obj.fly())
    print(obj.walk())
```

3. Полиморфизм: фигуры с методом area

Создай базовый класс Shape с методом area(), который возвращает 0. Создай классы-наследники:

Square (квадрат) – принимает сторону, площадь = сторона².

Circle (круг) – принимает радиус, площадь = π * r², используй π ≈ 3.14.

Triangle (треугольник) – принимает основание и высоту, площадь = 0.5 * основание * высота.

Переопредели метод area() в каждом классе, чтобы он возвращал площадь, округлённую до целого (используй round()). Считай с клавиатуры название фигуры (Square, Circle или Triangle), затем на следующей строке – необходимые параметры (одно или два целых числа через пробел). Создай объект и выведи результат вызова area().

Входные данные:

Две строки: название фигуры и параметры.

Выходные данные:

Одно целое число – площадь.

Пример ввода:

Square

5

Пример вывода:

25

Пример ввода 2:

Circle

3

Пример вывода 2:

28

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:

```
class Shape:
    def area(self):
        return 0

class Square(Shape):
    def __init__(self, side):
        self.side = side
    def area(self):
        return round(self.side ** 2)

class Circle(Shape):
    def __init__(self, radius):
        self.radius = radius
    def area(self):
        return round(3.14 * self.radius ** 2)

class Triangle(Shape):
    def __init__(self, base, height):
        self.base = base
        self.height = height
    def area(self):
        return round(0.5 * self.base * self.height)

s = input()
if s == 'Square':
    a = int(input())
    fig = Square(a)
elif s == 'Circle':
    r = int(input())
    fig = Circle(r)
elif s == 'Triangle':
    b, h = map(int, input().split())
    fig = Triangle(b, h)
print(fig.area())
```

4. Инкапсуляция: класс BankAccount (банковский счёт)

Создай класс BankAccount. В конструкторе принимается начальный баланс (число). Сохрани его в приватный атрибут __balance. Реализуй методы:

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

deposit(amount) – увеличивает баланс на amount (если amount положительное, иначе выводит «Ошибка»).
withdraw(amount) – уменьшает баланс на amount, если средств достаточно, иначе выводит «Недостаточно средств».
get_balance() – возвращает текущий баланс.
Сделай атрибут balance доступным через свойство @property (геттер), которое возвращает текущий баланс, и сеттер, который запрещает прямое изменение (при попытке присвоить выводит «Изменение баланса запрещено»). Считай с клавиатуры команды (ввод произвольный, в тестах будет проверка через eval), например: создание счёта, депозит, снятие, вывод баланса. Для удобства используй eval(input()) для выполнения выражений, как в практике.
Входные данные:
Строка с выражением (например, [acc := BankAccount(100), acc.deposit(50), acc.withdraw(30), acc.get_balance()]).
Выходные данные:
Результаты вызовов методов (выводятся через print внутри методов или возврат значения).
Пример ввода:
[acc := BankAccount(100), acc.deposit(50), acc.withdraw(30), acc.balance]
Пример вывода:
120
Решение:

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

    @property
    def balance(self):
        return self.__balance

    @balance.setter
    def balance(self, value):
        print('Изменение баланса запрещено')

    def deposit(self, amount):
        if amount > 0:
            self.__balance += amount
        else:
            print('Ошибка')

    def withdraw(self, amount):
        if amount > self.__balance:
            print('Недостаточно средств')
        else:
            self.__balance -= amount
```

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 get_balance(self):
	  return self.__balance
	
	eval(input())
	5. Инкапсуляция через setattr: класс Book
	Создай класс Book с атрибутом title (название). Используй механизм __setattr__, чтобы при попытке установить атрибут title проверять, что это строка длиной не более 50 символов. Если условие нарушено, выводить сообщение «Недопустимое название» и не изменять атрибут (оставить предыдущее значение, а если его не было, установить пустую строку). При попытке установить любой другой атрибут – разрешать без проверки. Также переопредели __getattribute__, чтобы при получении атрибута title возвращать его, а при получении любого другого – вести себя стандартно.
	Считай с клавиатуры две строки: первая – название книги, вторая – новое название (для проверки изменения). Создай объект с первым названием, затем попробуй изменить название на второе. Выведи финальное значение title.
	Входные данные:
	Две строки: название книги и новое название.
	Выходные данные:
	Сообщения об ошибках (если есть) и финальное название.
	Пример ввода:
	Война и мир
	Очень длинное название книги, которое превышает пятьдесят символов
	Пример вывода:
	Недопустимое название
	Война и мир
	Решение:
	class Book:
	  def __init__(self, title):
	  self.title = title
	
	  def __setattr__(self, name, value):
	  if name == 'title':
	  if isinstance(value, str) and len(value) <= 50:
	  super().__setattr__(name, value)
	  else:
	  print('Недопустимое название')
	  if not hasattr(self, 'title'):
	  super().__setattr__(name, '')
	  else:
	  super().__setattr__(name, value)
	
	  def __getattribute__(self, name):
	  return super().__getattribute__(name)
	
	title1 = input()

	title2 = input() book = Book(title1) book.title = title2 print(book.title)
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)

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	1. Калькулятор с обработкой ошибок и собственным исключением
	Напиши программу, которая запрашивает у пользователя два числа (целых или вещественных) и операцию: +, -, *, /, ** (возведение в степень). Программа должна вывести результат операции. Если ввод не является числом, вывести «Ошибка ввода». Если введена недопустимая операция, вывести «Неверная операция». Если происходит деление на ноль или другая ошибка (например, возведение отрицательного числа в дробную степень), вывести «Невозможно выполнить операцию». Для обработки неверной операции создай собственное исключение OperationError.
	Входные данные:
	Три строки: первое число, второе число, операция (каждая на новой строке).
	Выходные данные:
	Результат операции (число) или одно из сообщений об ошибке.
	Пример ввода 1:
	4
	2
	**
	Пример вывода 1:
	16.0
	Пример ввода 2:
	5
	0
	/
	Пример вывода 2:
	Невозможно выполнить операцию
	Пример ввода 3:
	a
	2
	+
	Пример вывода 3:
	Ошибка ввода
	Решение:
	class OperationError(Exception):
	  pass
	
	allowed_ops = '+-*/**'
	
	try:
	  a = float(input())
	  b = float(input())
	  op = input()
	  if op not in allowed_ops:
	  raise OperationError
	  if op == '+':

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

```
        result = a + b
    elif op == '-':
        result = a - b
    elif op == '*':
        result = a * b
    elif op == '/':
        result = a / b
    elif op == '**':
        result = a ** b
except ValueError:
    print('Ошибка ввода')
except OperationError:
    print('Неверная операция')
except Exception:
    print('Невозможно выполнить операцию')
else:
    print(result)
```

2. Класс Password с проверкой длины и наличия цифр

Создай класс Password, который при инициализации принимает строку с паролем и проверяет его корректность. Пароль считается корректным, если его длина не менее 6 символов и он содержит хотя бы одну цифру. Если пароль не проходит проверку, должны вызываться собственные исключения: MinLengthError (если длина меньше 6) или NoDigitError (если нет цифры). Проверка длины выполняется первой. Если пароль корректен, сохрани его в атрибуте password. Считай с клавиатуры строку, создай объект Password и выведи «Пароль: {password}» в случае успеха, иначе выведи сообщение «Короткий пароль» или «Нет цифры».

Входные данные:

Одна строка – пароль.

Выходные данные:

Строка с паролем или сообщение об ошибке.

Пример ввода 1:

abc123

Пример вывода 1:

Пароль: abc123

Пример ввода 2:

abcde

Пример вывода 2:

Короткий пароль

Пример ввода 3:

abcdefgh

Пример вывода 3:

Нет цифры

Решение:

```
class MinLengthError(Exception):
```

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

```
    pass

class NoDigitError(Exception):
    pass

class Password:
    def __init__(self, password):
        if len(password) < 6:
            raise MinLengthError('Короткий пароль')
        if not any(ch.isdigit() for ch in password):
            raise NoDigitError('Нет цифры')
        self.password = password

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

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

3. Класс TelephoneNumber с проверкой формата (11 цифр, начинается с 8)
Создай класс TelephoneNumber, который принимает строку с номером телефона. Корректный номер должен состоять ровно из 11 цифр и начинаться с цифры 8. Если длина не равна 11, вызови исключение LengthError. Если номер не начинается с 8, вызови исключение StartError. Если в номере есть нецифровые символы, вызови исключение NotDigitError. В случае успеха сохрани номер в атрибуте number. Считай с клавиатуры строку и создай объект. Если объект создан, выведи «Номер: {number}», иначе выведи имя класса исключения (например, LengthError). Для вывода имени используй print(e.__class__.__name__).
Входные данные:
Одна строка – номер телефона.
Выходные данные:
«Номер: ...» или имя исключения.
Пример ввода 1:
81234567890
Пример вывода 1:
Номер: 81234567890
Пример ввода 2:
1234567890
Пример вывода 2:

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

LengthError
Пример ввода 3:
8123456789a
Пример вывода 3:
NotDigitError
Решение:

```
class LengthError(Exception):
    pass

class StartError(Exception):
    pass

class NotDigitError(Exception):
    pass

class TelephoneNumber:
    def __init__(self, number):
        if len(number) != 11:
            raise LengthError
        if not number.startswith('8'):
            raise StartError
        if not number.isdigit():
            raise NotDigitError
        self.number = number

    def __str__(self):
        return f'Номер: {self.number}'

s = input()
try:
    tel = TelephoneNumber(s)
    print(tel)
except Exception as e:
    print(e.__class__.__name__)
```

4. Класс DefaultList (возвращает 0 при отсутствии элемента)
Создай класс DefaultList, унаследованный от list. При инициализации он принимает параметр with_exception (по умолчанию True). Если with_exception=False, то при обращении по индексу к несуществующему элементу должно возвращаться число 0 (вместо генерации IndexError). Если with_exception=True, поведение должно быть как у обычного списка (генерировать IndexError). Переопредели метод __getitem__. Для проверки используй следующий код (вставь его в программу после определения класса):

```
try:
```

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

```
    eval(input())
except Exception as e:
    print(e.__class__.__name__)
```

Входные данные:

Строка с выражением, создающим объект и обращающимся к элементу.

Выходные данные:

Результат выражения или имя исключения.

Пример ввода 1:

```
[lst := DefaultList(with_exception=False), lst.append(1), lst.append(2), print(lst[0]), print(lst[1]), print(lst[2])]
```

Пример вывода 1:

```
1
2
0
```

Пример ввода 2:

```
[lst := DefaultList(with_exception=True), lst.append(1), print(lst[1])]
```

Пример вывода 2:

```
IndexError
```

Решение:

```
class DefaultList(list):
    def __init__(self, with_exception=True):
        super().__init__()
        self.with_exception = with_exception

    def __getitem__(self, index):
        if self.with_exception:
            return super().__getitem__(index)
        else:
            try:
                return super().__getitem__(index)
            except IndexError:
                return 0

try:
    eval(input())
except Exception as e:
    print(e.__class__.__name__)
```

5. Класс User с проверкой возраста и имени

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 User, который при инициализации принимает имя (строка) и возраст (целое число). Проверь, что имя не пустое (длина > 0) и возраст не менее 18 лет. Если имя пустое, вызови исключение NameError с сообщением «Имя не может быть пустым». Если возраст меньше 18, вызови исключение AgeError с сообщением «Возраст должен быть не менее 18». Если данные корректны, сохрани их в атрибутах name и age. Считай с клавиатуры имя и возраст (каждое на новой строке), создай объект User и выведи «Пользователь: {name}, возраст: {age}». В случае ошибки выведи соответствующее сообщение.
	Входные данные:
	Имя (строка), возраст (целое число) – каждое на отдельной строке.
	Выходные данные:
	Информация о пользователе или сообщение об ошибке.
	Пример ввода 1:
	Анна
	25
	Пример вывода 1:
	Пользователь: Анна, возраст: 25
	Пример ввода 2:
	17
	Пример вывода 2:
	Возраст должен быть не менее 18
	Пример ввода 3:
	
	25
	Пример вывода 3:
	Имя не может быть пустым
	Решение:
	class NameError(Exception):
	  pass
	
	class AgeError(Exception):
	  pass
	
	class User:
	  def __init__(self, name, age):
	  if not name:
	  raise NameError('Имя не может быть пустым')
	  if age < 18:
	  raise AgeError('Возраст должен быть не менее 18')
	  self.name = name
	  self.age = age
	
	  def __str__(self):
	  return f'Пользователь: {self.name}, возраст: {self.age}'

	try:
	  name = input()
	  age = int(input())
	  user = User(name, age)
	  print(user)
	except NameError as e:
	  print(e)
	except AgeError as e:
	  print(e)
	except ValueError:
	  print('Ошибка ввода возраста')

Модуль 3, 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
владелец Маршина Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 5fba31f1c84f2e80fe6f27ace01aeaa0
срок действия 05.12.2025 - 05.03.2027

Варианты заданий	1. Калькулятор с аргументами командной строки (argparse)
	Напиши программу, которая принимает два целых числа как позиционные аргументы и необязательный ключ -op с выбором операции: add, sub, mul, div (по умолчанию add). Также добавь ключ --round (флаг), который округляет результат до целого (если указан). При делении на ноль выводить «Error». Используй модуль argparse.
	Примеры работы:
	$ python calc.py 10 3
	13
	$ python calc.py 10 3 -op mul
	30
	$ python calc.py 10 3 -op div --round
	3
	$ python calc.py 10 0 -op div
	Error
	Входные данные:
	Аргументы командной строки.
	Выходные данные:
	Число или «Error».
	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('-op', choices=['add','sub','mul','div'], default='add')
	parser.add_argument('--round', action='store_true')
	args = parser.parse_args()
	
	try:
	res = ops[args.op](args.a, args.b)
	if args.round:
	print(round(res))
	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(res)
except ZeroDivisionError:
    print('Error')
```

2. Количество дней до дня рождения

Напиши программу, которая запрашивает у пользователя дату рождения в формате ДД.ММ.ГГГГ и выводит количество полных лет и количество дней до следующего дня рождения (в днях). Если день рождения уже прошёл в текущем году, считается следующий год. Используй модуль datetime. Вывод оформи так: «Вам N лет. До следующего дня рождения осталось D дней.»

Входные данные:

Одна строка – дата рождения.

Выходные данные:

Одна строка с количеством лет и дней.

Пример ввода (сегодня 01.01.2026):

15.03.2000

Пример вывода:

Вам 25 лет. До следующего дня рождения осталось 73 дней.

Решение:

```
from datetime import datetime, date

birth = datetime.strptime(input(), '%d.%m.%Y').date()
today = date.today()
age = today.year - birth.year
if (today.month, today.day) < (birth.month, birth.day):
    age -= 1
# следующий день рождения
next_birth = date(today.year, birth.month, birth.day)
if next_birth < today:
    next_birth = date(today.year + 1, birth.month, birth.day)
delta = (next_birth - today).days
print(f'Вам {age} лет. До следующего дня рождения осталось {delta} дней.')
```

3. Поиск всех email-адресов в тексте

Напиши программу, которая читает строку текста и выводит все найденные email-адреса (каждый на новой строке) в порядке их появления. Email считается корректным, если он содержит @, до и после которого есть как минимум один символ из букв, цифр, точки или дефиса. Используй модуль re.

Входные данные:

Одна строка текста.

Выходные данные:

Все email-адреса, каждый на новой строке.

Пример ввода:

Контакты: admin@site.com, support@mail.ru, и test@yandex.ru

Пример вывода:

admin@site.com

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

support@mail.ru
test@yandex.ru
Решение:

```
import re
text = input()
emails = re.findall(r'[\w.-]+@[\w.-]+', text)
for email in emails:
    print(email)
```

4. Функция с doctest и unittest для нахождения уникальных чисел
Напиши функцию unique_numbers(lst), которая принимает список чисел и возвращает список чисел, встречающихся ровно один раз, в порядке их первого появления. Добавь докстринг с примерами (doctest). Также напиши unittest-тесты для этой функции (в отдельном файле или том же). В основной программе считай список чисел с клавиатуры и выведи результат через пробел.
Входные данные:
Строка чисел через пробел.
Выходные данные:
Числа, встречающиеся один раз, через пробел.
Пример ввода:
1 2 2 3 4 4 5
Пример вывода:
1 3 5
Решение:

```
def unique_numbers(lst):
    """
    >>> unique_numbers([1, 2, 2, 3, 4, 4, 5])
    [1, 3, 5]
    >>> unique_numbers([1, 1, 2])
    [2]
    >>> unique_numbers([1, 2, 3])
    [1, 2, 3]
    """
    from collections import Counter
    counts = Counter(lst)
    return [x for x in lst if counts[x] == 1]

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

# unittest в отдельном файле (test_unique.py):
```

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 unittest
	from main import unique_numbers
	class TestUnique(unittest.TestCase):
	  def test_mixed(self):
	  self.assertEqual(unique_numbers([1,2,2,3,4,4,5]), [1,3,5])
	  def test_empty(self):
	  self.assertEqual(unique_numbers([]), [])
	  def test_all_unique(self):
	  self.assertEqual(unique_numbers([5,4,3]), [5,4,3])
	if __name__ == '__main__':
	  unittest.main()
	"""
	В основной программе:
	lst = list(map(int, input().split()))
	print(*unique_numbers(lst))
	5. Система логирования попыток входа
	Напиши программу, которая имитирует систему входа с проверкой доступа. В программе задан словарь users = {"admin": "1234", "user": "qwerty"}. Программа запрашивает логин и пароль (по отдельным строкам). Если логин не найден, выводится «Пользователь не найден» и в лог записывается ошибка уровня ERROR. Если пароль неверен, выводится «Неверный пароль» и в лог – предупреждение WARNING. Если вход успешен, выводится «Добро пожаловать» и в лог – INFO. Логи должны записываться в файл login.log с форматом: %(asctime)s - %(levelname)s - %(message)s. Обработай возможные исключения с try/except и запиши их в лог с logging.exception().
	Входные данные:
	Две строки: логин и пароль.
	Выходные данные:
	Сообщение о результате.
	Пример ввода:
	admin
	1234
	Пример вывода:
	Добро пожаловать
	Решение:
	import logging
	
	logging.basicConfig(
	  level=logging.INFO,
	  filename='login.log',
	  encoding='utf-8',
	  format='%(asctime)s - %(levelname)s - %(message)s'
	)

	users = {"admin": "1234", "user": "qwerty"}
	
	try:
	  login = input()
	  password = input()
	  if login not in users:
	  print('Пользователь не найден')
	  logging.error(f'Попытка входа неизвестного пользователя: {login}')
	  elif users[login] != password:
	  print('Неверный пароль')
	  logging.warning(f'Неверный пароль для пользователя: {login}')
	  else:
	  print('Добро пожаловать')
	  logging.info(f'Успешный вход пользователя: {login}')
	except Exception:
	  logging.exception('Ошибка при выполнении программы')

	Модуль 3, WEB и API, Ответственное использование ИИ-инструментов
Количество академических часов	
Формы контроля	
Диагностические инструменты	
Показатели и критерии оценивания	
	Модуль 3, WEB и API, Использование библиотек

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

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. Калькулятор с выбором операции (argparse)
	Напиши программу calc.py, которая принимает два целых числа (позиционные аргументы) и опциональный ключ -o (или --operation) с возможными значениями: add, sub, mul, pow, mod.
	add – вывести сумму.
	sub – разность (первое минус второе).
	mul – произведение.
	pow – первое число, возведённое в степень второго.
	mod – остаток от деления первого на второе.
	Если ключ не указан, по умолчанию выполняется сложение.
	В случае ошибки (например, деление на ноль) вывести строку «Ошибка».
	Примеры вызова и вывода
	$ python calc.py 5 3
	8
	
	$ python calc.py 5 3 -o mul
	15
	
	$ python calc.py 2 3 -o pow
	8
	
	$ python calc.py 5 0 -o mod
	Ошибка
	Решение
	import argparse
	
	ops = {
	'add': lambda a, b: a + b,
	'sub': lambda a, b: a - b,
	'mul': lambda a, b: a * b,
	'pow': lambda a, b: a ** b,
	'mod': lambda a, b: a % b
	}
	
	parser = argparse.ArgumentParser()
	parser.add_argument('a', type=int)
	parser.add_argument('b', type=int)
	parser.add_argument('-o', '--operation', 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.operation](args.a, args.b))
except Exception:
    print('Ошибка')
```

Задача 2. Расчёт возраста по дате рождения (datetime)

Напиши программу, которая запрашивает дату рождения в формате ДД.ММ.ГГГГ и выводит количество полных лет, исполнившихся на текущую дату (используй datetime.date.today()). Если день рождения ещё не наступил в текущем году, вычитай один год.

Входные данные

Строка с датой в формате ДД.ММ.ГГГГ.

Выходные данные

Одно целое число – количество полных лет.

Примеры

(Если сегодня 15.06.2026)

Ввод:

10.06.2000

Вывод:

26

Ввод:

20.06.2000

Вывод:

25

Примечание

Подсказки внутри input() не используй.

Решение

```
import datetime

birth = datetime.datetime.strptime(input(), '%d.%m.%Y').date()
today = datetime.date.today()
age = today.year - birth.year
if (today.month, today.day) < (birth.month, birth.day):
    age -= 1
print(age)
```

Задача 3. Поиск email-адресов в тексте (регулярные выражения)

Напиши программу, которая находит в заданном тексте все адреса электронной почты в формате имя@домен.зона, где:

имя – латинские буквы, цифры, точки, дефисы, нижние подчёркивания;

домен – латинские буквы и цифры;

зона – от 2 до 4 латинских букв.

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

Выведи все найденные адреса в порядке появления, каждый на новой строке. Если адресов нет, выведи «Не найдено».
Входные данные
Одна строка с текстом.
Выходные данные
Список email-адресов или сообщение.
Пример
Ввод:
Пишите на почту: info@company.ru или support@mail.com. Также можно связаться через form@site.net
Вывод:
info@company.ru
support@mail.com
form@site.net
Примечание
Подсказки внутри input() не используй.
Решение
:
```
import re

text = input()
pattern = r'[a-zA-Z0-9._-]+@[a-zA-Z0-9-]+\.[a-zA-Z]{2,4}'
emails = re.findall(pattern, text)
if emails:
    print('\n'.join(emails))
else:
    print('Не найдено')
```
Задача 4. Поиск и исправление ошибок в коде (ИИ-инструменты)
Ниже дан код, который должен подсчитывать, сколько раз в строке встречается подстрока «Python» (без учёта регистра). Однако код содержит ошибку. Найди её и исправь.
Исходный код с ошибкой
```
text = input()
count = 0
for i in range(len(text) - len('Python')):
    if text[i:i+len('Python')].lower() == 'python':
        count += 1
print(count)
```
Входные данные
Одна строка.
Выходные данные
Одно целое число – количество вхождений.

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
	Ввод:
	Python and python and PYTHON
	Вывод:
	3
	Решение (исправленный код)
	text = input()
	count = 0
	sub = 'Python'
	for i in range(len(text) - len(sub) + 1): # исправлено: +1
	  if text[i:i+len(sub)].lower() == sub.lower():
	  count += 1
	print(count)
	Задача 5. Генератор последовательности чисел (argparse)
	Напиши программу seq.py, которая генерирует последовательность целых чисел. Параметры задаются через командную строку:
	-start – начало последовательности (по умолчанию 1).
	-end – конец последовательности включительно (по умолчанию 20).
	-step – шаг (по умолчанию 1).
	Программа должна вывести все числа из диапазона [start, end] с заданным шагом, каждое на новой строке.
	Примеры вызова и вывода
	$ python seq.py -start 3 -end 10 -step 2
	3
	5
	7
	9
	
	$ python seq.py -start 10 -end 1 -step -2
	10
	8
	6
	4
	2
	Решение
	import argparse
	
	parser = argparse.ArgumentParser()
	parser.add_argument('-start', type=int, default=1)

	parser.add_argument('-end', type=int, default=20)
	parser.add_argument('-step', type=int, default=1)
	args = parser.parse_args()
	
	num = args.start
	if args.step > 0:
	  while num <= args.end:
	  print(num)
	  num += args.step
	else:
	  while num >= args.end:
	  print(num)
	  num += args.step

Модуль 3, 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 (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 Мариния Виктория Вадимовна
серийный номер 01A91D1C547E0D7F0E0F
срок действия 05.12.2025 - 05.03.2027

Варианты заданий	Задача 1. Количество вакансий разработчиков (API «Работа России») Напиши программу, которая делает запрос к API «Работа России» по адресу: http://opendata.trudvsem.ru/api/v1/vacancies/region/77 (регион 77 – Москва) с параметром text равным «разработчик». Из полученного JSON-ответа извлеки количество вакансий (поле results.vacancies – список, количество элементов = ответ). Выведи полученное число. Входные данные Нет (регион и текст запроса фиксированы). Выходные данные Одно целое число – количество найденных вакансий. Пример вывода 42 Решение import requests url = 'http://opendata.trudvsem.ru/api/v1/vacancies/region/77' params = {'text': 'разработчик'} response = requests.get(url, params=params) data = response.json() vacancies = data['results']['vacancies'] print(len(vacancies)) Задача 2. Подсчёт количества PEP в секции «Accepted» Напиши программу, которая парсит страницу https://peps.python.org/ и подсчитывает количество PEP в секции с идентификатором accepted-peps. Внутри этой секции все PEP перечислены в таблице. Подсчитай количество строк <tr> в таблице и выведи число. Входные данные Нет. Выходные данные Одно целое число. Пример вывода 9 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='accepted-peps')

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

```
table = section.find('table')
rows = table.find_all('tr')
print(len(rows))
```

Задача 3. Модульное тестирование функции «проверка чётности» (unittest)

Напиши функцию is_even(n), которая принимает целое число и возвращает True, если число чётное, и False в противном случае.

Затем создай класс тестов TestIsEven (наследник unittest.TestCase) с тремя методами:

test_even_numbers – проверяет, что числа 2, 4, 0 возвращают True;

test_odd_numbers – проверяет, что числа 1, 3, -5 возвращают False;

test_negative_even – проверяет, что отрицательные чётные числа (например, -4) возвращают True.

В основной программе прочитай одно число, вызови функцию и выведи результат. Тесты должны запускаться автоматически (используй unittest.main(argv=[''], exit=False)).

Входные данные

Одно целое число.

Выходные данные

True или False.

Пример

Ввод:

```
4
```

Вывод:

```
True
```

Решение

```
import unittest

def is_even(n):
    return n % 2 == 0

class TestIsEven(unittest.TestCase):
    def test_even_numbers(self):
        self.assertTrue(is_even(2))
        self.assertTrue(is_even(4))
        self.assertTrue(is_even(0))
    def test_odd_numbers(self):
        self.assertFalse(is_even(1))
        self.assertFalse(is_even(3))
        self.assertFalse(is_even(-5))
    def test_negative_even(self):
        self.assertTrue(is_even(-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

```
if __name__ == '__main__':
    # Запуск тестов
    unittest.main(argv=[''], exit=False)
    # Основная программа
    n = int(input())
    print(is_even(n))
```

Задача 4. Git и README: выбор правильных утверждений

Ответь на вопросы и выведи фрагмент README.

Вопрос 4.1. Какое расширение должен иметь файл README с описанием проекта на языке Markdown?

Выбери один верный вариант ответа:

.txt

.md

.readme

.docx

Вопрос 4.2. Какие разделы должен содержать качественный файл README.md? (Выбери все верные варианты.)

Название проекта

Пароли и токены

Описание проекта

Технологии, которые были использованы

Инструкцию по запуску проекта

Информацию об авторе

Вопрос 4.3. Как правильно оформить ссылку в Markdown?

<https://vk.ru>

[vk.ru] https://vk.ru

vk.ru

(vk.ru)[https://vk.ru]

Вопрос 4.4. Напиши минимальный фрагмент README.md для проекта «Парсер вакансий с API». Фрагмент должен содержать:

заголовок первого уровня с названием проекта;

подзаголовок второго уровня «Описание» и краткое описание (одно предложение);

подзаголовок второго уровня «Технологии» и список (маркированный) из двух технологий: Python, requests.

Выведи этот фрагмент на экран (как строку).

Входные данные

Нет.

Выходные данные

Многострочная строка с Markdown-разметкой.

Пример вывода

```
# Парсер вакансий с API
```

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
	Программа получает список вакансий через API.
	
	## Технологии
	- Python
	- requests
	Решение (для всех четырёх подзадач)
	# Ответы на вопросы
	print('.md')
	print('Название проекта, Описание проекта, Технологии, Инструкцию по запуску, Информацию об авторе')
	print('vk.ru')
	# Вывод фрагмента README
	readme = """# Парсер вакансий с API
	
	## Описание
	Программа получает список вакансий через API.
	
	## Технологии
	- Python
	- requests"""
	print(readme)
	Задача 5. Город по IP-адресу (API DaData)
	Напиши программу, которая использует API DaData «Город по IP-адресу» (https://dadata.ru/api/iplocate/). Для работы требуется токен (в учебных целях используй тестовый токен "test_token"). Программа запрашивает IP-адрес, отправляет POST-запрос к API и выводит 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
	Ввод:
	46.226.227.20
	Вывод:
	г Москва
	Решение
	import requests
	import json

	token = "test_token" # В реальной задаче токен должен быть получен
	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")

Модуль 3, WEB и API, Проект	
Количество академических часов	
Формы контроля	
Диагностические инструменты	
Показатели и критерии оценивания	
Модуль 4, Базы данных, PyQT	
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 01ba2101e94f2cb04c6f27cc019aedad
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. Номеронабиратель (интерфейс)
	Создай окно с кнопками цифр и символов * и #, расположенными в виде телефонной клавиатуры (4 строки по 3 кнопки) с использованием QGridLayout. Кнопки должны иметь размер 80×80 пикселей и шрифт 18 пунктов. Функционал кнопок не требуется.
	Входные данные:
	Нет.
	Выходные данные:
	Графическое окно с кнопками.
	Пример:
	На экране появляется окно с кнопками:
	1 2 3
	4 5 6
	7 8 9
	* 0 #
	Решение:
	import sys
	from PyQt6.QtWidgets import QApplication, QWidget, QGridLayout, QPushButton
	from PyQt6.QtGui import QFont
	
	class Dialer(QWidget):
	def __init__(self):
	super().__init__()
	self.setWindowTitle("Номеронабиратель")
	layout = QGridLayout()
	keys = [['1','2','3'], ['4','5','6'], ['7','8','9'], ['*','0','#']]
	for i in range(4):
	for j in range(3):
	btn = QPushButton(keys[i][j])
	btn.setFixedSize(80, 80)
	btn.setFont(QFont("Arial", 18))
	layout.addWidget(btn, i, j)
	self.setLayout(layout)
	
	if __name__ == "__main__":
	app = QApplication(sys.argv)
	w = Dialer()
	w.show()
	sys.exit(app.exec())
	2. Отображение выбранного месяца

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 программу с выпадающим списком (QComboBox) месяцев. При выборе любого месяца в метке (QLabel) сразу отображается выбранный месяц (без кнопки). Используй QVBoxLayout.
Входные данные:
Выбор пункта в списке.
Выходные данные:
Текст в метке, например, «Выбран: Январь».
Пример:
Пользователь выбирает «Март» → на экране появляется «Выбран: Март».
Решение:

```
import sys
from PyQt6.QtWidgets import QApplication, QWidget, QVBoxLayout, QComboBox, QLabel

class MonthPicker(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Выбор месяца")
        layout = QVBoxLayout()
        self.combo = QComboBox()
        self.combo.addItems(["Январь","Февраль","Март","Апрель","Май","Июнь",
                             "Июль","Август","Сентябрь","Октябрь","Ноябрь","Декабрь"])
        self.label = QLabel("Выберите месяц")
        self.combo.currentTextChanged.connect(self.label.setText)
        layout.addWidget(self.combo)
        layout.addWidget(self.label)
        self.setLayout(layout)

if __name__ == "__main__":
    app = QApplication(sys.argv)
    w = MonthPicker()
    w.show()
    sys.exit(app.exec())
```

3. Конвертер Цельсий → Фаренгейт (с QDoubleSpinBox)
Создай окно с числовым полем (QDoubleSpinBox) для ввода градусов Цельсия и меткой (QLabel) для вывода результата в Фаренгейтах. При изменении значения в поле автоматически пересчитывается и выводится результат с двумя знаками после запятой. Формула: F = C * 9/5 + 32.
Входные данные:
Число (градусы Цельсия) в QDoubleSpinBox.
Выходные данные:
Число (градусы Фаренгейта) в метке.
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

В поле введено 25 → на экране появляется 77.00 °F.
Решение:

```
import sys
from PyQt6.QtWidgets import QApplication, QWidget, QVBoxLayout, QDoubleSpinBox, QLabel

class CtoF(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Цельсий → Фаренгейт")
        layout = QVBoxLayout()
        self.spin = QDoubleSpinBox()
        self.spin.setRange(-100, 100)
        self.spin.setValue(0)
        self.label = QLabel("0.00 °F")
        self.spin.valueChanged.connect(self.convert)
        layout.addWidget(self.spin)
        layout.addWidget(self.label)
        self.setLayout(layout)

    def convert(self, c):
        f = c * 9/5 + 32
        self.label.setText(f"{f:.2f} °F")

if __name__ == "__main__":
    app = QApplication(sys.argv)
    w = CtoF()
    w.show()
    sys.exit(app.exec())
```

4. Генератор случайного пароля (фиксированная длина)
Создай программу с кнопкой «Сгенерировать». При нажатии генерируется случайный пароль длиной 8 символов из букв латинского алфавита (верхний и нижний регистр) и цифр, и выводится в поле QLineEdit (только для чтения). Используй модули random и string.
Входные данные:
Нажатие кнопки.
Выходные данные:
Строка с паролем в поле ввода.
Пример:
После нажатия кнопки в поле появляется, например, aB3dF8gH.
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 sys, random, string
	from PyQt6.QtWidgets import QApplication, QWidget, QVBoxLayout, QPushButton, QLineEdit
	
	class PasswordGen(QWidget):
	  def __init__(self):
	  super().__init__()
	  self.setWindowTitle("Генератор пароля")
	  layout = QVBoxLayout()
	  self.btn = QPushButton("Сгенерировать")
	  self.btn.clicked.connect(self.generate)
	  self.edit = QLineEdit()
	  self.edit.setReadOnly(True)
	  layout.addWidget(self.btn)
	  layout.addWidget(self.edit)
	  self.setLayout(layout)
	
	  def generate(self):
	  chars = string.ascii_letters + string.digits
	  self.edit.setText(''.join(random.choice(chars) for _ in range(8)))
	
	if __name__ == "__main__":
	  app = QApplication(sys.argv)
	  w = PasswordGen()
	  w.show()
	  sys.exit(app.exec())
	5. Форма авторизации (без проверки)
	Создай окно с полями «Логин» и «Пароль» (пароль скрыт) и кнопкой «Войти». При нажатии кнопки появляется всплывающее сообщение QMessageBox с текстом «Привет!». Используй QFormLayout.
	Входные данные:
	Нет (ввод не влияет на результат).
	Выходные данные:
	Окно с сообщением.
	Пример:
	После ввода любых данных и нажатия кнопки появляется окно с надписью «Привет!».
	Решение:

	import sys
	from PyQt6.QtWidgets import QApplication, QWidget, QFormLayout, QLineEdit, QPushButton, QMessageBox
	
	class Login(QWidget):
	  def __init__(self):
	  super().__init__()
	  self.setWindowTitle("Авторизация")
	  layout = QFormLayout()
	  self.login = QLineEdit()
	  self.password = QLineEdit()
	  self.password.setEchoMode(QLineEdit.EchoMode.Password)
	  btn = QPushButton("Войти")
	  btn.clicked.connect(lambda: QMessageBox.information(self, "", "Привет!"))
	  layout.addRow("Логин:", self.login)
	  layout.addRow("Пароль:", self.password)
	  layout.addRow(btn)
	  self.setLayout(layout)
	
	if __name__ == "__main__":
	  app = QApplication(sys.argv)
	  w = Login()
	  w.show()
	  sys.exit(app.exec())

Модуль 4, Базы данных, Базы 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

Контур Крипто
владелец Маршеня Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba2191e94f2eb04e6f27ee019aedad
срок действия с 05.12.2025 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	1. Фильмы, вышедшие до 1990 года, с сортировкой по убыванию года Напиши SQL-запрос, который выводит названия и годы выпуска фильмов, вышедших до 1990 года, отсортированных по году в порядке убывания (от новых к старым). Входные данные: Нет (запрос выполняется к базе данных). Выходные данные: Таблица с двумя колонками: title (название) и year (год). Пример вывода (часть): title year Терминатор 1984 Бегущий по лезвию 1982 Решение (SQL-запрос): SELECT title, year FROM movies WHERE year < 1990 ORDER BY year DESC; 2. Фильмы длительностью более 2 часов с именами режиссёров Напиши SQL-запрос, который выводит названия фильмов и имена их режиссёров для фильмов, длительность которых превышает 120 минут. Используй соединение таблиц (JOIN). Входные данные: Нет. Выходные данные: Таблица с колонками: title (название фильма) и name (имя режиссёра). Пример вывода: Властелин колец: Братство Кольца | Питер Джексон Тёмный рыцарь | Кристофер Нолан Решение (SQL-запрос): SELECT movies.title, directors.name FROM movies JOIN directors ON movies.director_id = directors.id WHERE movies.duration > 120; 3. Добавление нового режиссёра и фильма Напиши SQL-запросы для добавления нового режиссёра «Квентин Тарантино» и фильма «Криминальное чтиво» (год 1994, длительность 154 минуты), связав фильм с этим режиссёром (используй его id). Предположим, что после вставки режиссёра его id стал равен 7 (можно использовать подзапрос или явно указать). Входные данные: Нет (данные для вставки известны).

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: В таблицах directors и movies появляются новые строки. Пример: После выполнения запросов в таблице directors будет запись с name = 'Квентин Тарантино', а в movies – с title = 'Криминальное чтиво' и director_id = 7. Решение (SQL-запросы): -- Добавление режиссёра INSERT INTO directors(name) VALUES ('Квентин Тарантино'); -- Добавление фильма (указываем director_id = 7) INSERT INTO movies(title, year, duration, director_id) VALUES ('Криминальное чтиво', 1994, 154, 7); 4. Обновление длительности фильма «Терминатор» Напиши SQL-запрос, который изменяет длительность фильма «Терминатор» на 107 минут (изначально была 108 или иная). Входные данные: Название фильма и новая длительность заданы. Выходные данные: В таблице movies у записи с названием «Терминатор» обновляется поле duration. Пример: Было: Терминатор | 1984 | 108 → Стало: Терминатор | 1984 | 107 Решение (SQL-запрос): UPDATE movies SET duration = 107 WHERE title = 'Терминатор'; 5. Удаление фильма по названию Напиши SQL-запрос для удаления фильма «Криминальное чтиво» из таблицы movies (предполагается, что он был добавлен ранее). Удаление должно происходить по названию. Входные данные: Название фильма «Криминальное чтиво». Выходные данные: Из таблицы movies удаляется строка с указанным названием. Пример: После выполнения запроса фильм «Криминальное чтиво» перестаёт присутствовать в таблице. Решение (SQL-запрос): DELETE FROM movies WHERE title = 'Криминальное чтиво';
Модуль 4, Базы данных, Командный проект	
Количество академических часов	
Формы контроля	
Диагностические инструменты	
Показатели и критерии оценивания	
Модуль 4, Базы данных, 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.
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
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	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 два целых числа и выводит их произведение. Входные данные: Два целых числа, каждое на отдельной строке. Выходные данные: Одно целое число – произведение. Пример ввода: 7 8 Пример вывода: 56 Возможное решение: a = int(input()) b = int(input()) print(a * b) Задание 2 (написание кода) Напиши программу, которая запрашивает радиус круга (вещественное число) и вычисляет его площадь по формуле $S = \pi \cdot R^2$. Используй значение $\pi = 3.1415$. Результат выведи без округления. Входные данные: Одно вещественное число – радиус. Выходные данные: Одно вещественное число – площадь круга. Пример ввода: 4 Пример вывода: 50.264 Возможное решение: r = float(input()) pi = 3.1415 print(pi * r ** 2) Задание 3 (написание кода) Напиши программу, которая запрашивает целое трёхзначное число и выводит сумму его цифр. Входные данные: Одно целое трёхзначное число. Выходные данные: Одно целое число – сумма цифр. Пример ввода: 456

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:
15
Возможное решение:
```
n = int(input())
d1 = n // 100
d2 = n // 10 % 10
d3 = n % 10
print(d1 + d2 + d3)
```
Задание 4 (написание кода)
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.
Пример ввода:
12
7
9
Пример вывода:
12
Возможное 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)
```
Задание 5 (написание кода)
Напиши программу, которая запрашивает целое положительное число и определяет, является ли оно палиндромом (читается одинаково слева направо и справа налево). Программа должна вывести «Палиндром» или «Не палиндром».
Входные данные:
Одно целое положительное число.
Выходные данные:
Одна строка – «Палиндром» или «Не палиндром».
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

12321
Пример вывода:
Палиндром
Возможное решение:

```
n = input()
if n == n[::-1]:
    print('Палиндром')
else:
    print('Не палиндром')
```

Задание 6 (написание кода)
Напиши программу, которая запрашивает 6 целых чисел (каждое на отдельной строке) и формирует два списка: список чётных чисел (в порядке ввода) и список нечётных чисел (в порядке ввода). Выведи оба списка: сначала чётные, затем нечётные (каждый список на отдельной строке в формате Python).
Входные данные:
6 целых чисел (по одному на строке).
Выходные данные:
Две строки: список чётных и список нечётных чисел.
Пример ввода:
3
8
1
6
2
5
Пример вывода:
[8, 6, 2]
[3, 1, 5]
Возможное решение:

```
evens = []
odds = []
for _ in range(6):
    x = int(input())
    if x % 2 == 0:
        evens.append(x)
    else:
        odds.append(x)
print(evens)
print(odds)
```

Задание 7 (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 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 ввода:
яблоко банан апельсин банан груша банан
Пример вывода:
банан
Возможное 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
```

Задание 8 (написание кода)
Напиши функцию filter_numbers(*args, kind), которая принимает произвольное количество целых чисел (позиционные аргументы) и один именованный аргумент kind, который может быть равен 'odd' или 'even'. Функция должна возвращать список нечётных чисел, если kind='odd', и список чётных чисел, если kind='even'. В основной программе прочитай строку целых чисел через пробел, затем строку с видом фильтрации, вызови функцию и выведи результат.
Входные данные:
Первая строка – целые числа через пробел. Вторая строка – 'odd' или 'even'.
Выходные данные:
Список чисел.
Пример ввода:

```
10 15 20 25 30
odd
```

Пример вывода:

```
[15, 25]
```

Возможное решение:

```
def filter_numbers(*args, kind):
    if kind == 'odd':
        return [x for x in args if x % 2 != 0]
    else:
        return [x for x in args 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

```
numbers = list(map(int, input().split()))
kind = input()
print(filter_numbers(*numbers, kind=kind))
```

Задание 9 (исправление ошибок)

Дана программа, которая должна запрашивать целое число и выводить «Число положительное», если оно больше нуля, иначе «Число не положительное»:

```
n = input()
if n > 0
    print('Число положительное')
else
    print('Число не положительное')
```

Программа содержит синтаксические ошибки. Найди и исправь их.

Входные данные:

Одно целое число.

Выходные данные:

Одна строка – «Число положительное» или «Число не положительное».

Пример ввода:

```
-5
```

Пример вывода:

```
Число не положительное
```

Возможное решение:

```
n = int(input())
if n > 0:
    print('Число положительное')
else:
    print('Число не положительное')
```

Задание 10 (исправление ошибок)

Дана программа, которая должна выводить числа от 1 до n (включительно) с шагом 2, где n вводится с клавиатуры:

```
n = input()
for i in range(0, n, 2)
print(i)
```

Программа содержит ошибки. Найди и исправь их.

Входные данные:

Одно целое число n.

Выходные данные:

Нечётные числа от 1 до n, каждое на отдельной строке.

Пример ввода:

```
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:
1
3
5
7
Возможное решение:

```
n = int(input())
for i in range(1, n + 1, 2):
print(i)
```

Задание 11 (исправление ошибок)
Дана программа, которая должна вычислять сумму цифр введённого числа:

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

Программа содержит логическую ошибку. Найди и исправь её.
Входные данные:
Одно целое положительное число.
Выходные данные:
Одно целое число – сумма цифр.
Пример ввода:
1234
Пример вывода:
10
Возможное решение:

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

Задание 12 (исправление ошибок)
Дана программа, которая должна проверять, является ли введённый год високосным. Год является високосным, если он кратен 4, но не кратен 100, или кратен 400:

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

	year = int(input())
	if year % 4 == 0:
	  print('Високосный')
	elif year % 100 != 0:
	  print('Високосный')
	elif year % 400 == 0:
	  print('Високосный')
	else:
	  print('Обычный')
	Программа содержит логическую ошибку. Найди и исправь её.
	Входные данные:
	Одно целое число – год.
	Выходные данные:
	Одна строка – «Високосный» или «Обычный».
	Пример ввода:
	1900
	Пример вывода:
	Обычный
	Возможное решение:
	year = int(input())
	if (year % 4 == 0 and year % 100 != 0) or year % 400 == 0:
	  print('Високосный')
	else:
	  print('Обычный')
	Задание 13 (краткий ответ)
	Вопрос: Какой оператор используется для возврата значения из функции в Python? Запиши одним словом.
	Ответ: return
	Задание 14 (краткий ответ)
	Вопрос: Как называется документ, описывающий стиль написания кода на Python? В ответе укажи название, состоящее из буквы, цифры и пробела.
	Ответ: PEP 8
	Задание 15 (краткий ответ)
	Вопрос: Какой встроенной функции нет в Python: print(), input(), sqrt() или len()? Запиши название функции.
	Ответ: sqrt

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 Film с атрибутами: title (название), director (режиссёр), year (год выпуска). Реализуй конструктор __init__ и метод info(), который выводит строку в формате:
	«{режиссёр} — {название} ({год})».
	В основной программе прочитай три строки (название, режиссёр, год), создай объект и вызови метод info().
	Входные данные:
	Три строки: название, режиссёр, год (целое число).
	Выходные данные:
	Одна строка.
	Пример ввода:
	Начало
	Кристофер Нолан
	2010
	Пример вывода:
	Кристофер Нолан — Начало (2010)
	Возможное решение:
	class Film:
	  def __init__(self, title, director, year):
	  self.title = title
	  self.director = director
	  self.year = year
	  def info(self):
	  print(f'{self.director} — {self.title} ({self.year})')
	
	title = input()
	director = input()
	year = int(input())
	film = Film(title, director, year)
	film.info()
	Задание 2 (написание кода)
	Создай класс Employee с атрибутами name (имя) и salary (зарплата – целое число). Реализуй:
	конструктор, который проверяет, что зарплата положительная; если нет – установи salary = 0 и выведи сообщение «Зарплата исправлена на 0»;
	метод bonus(percent), который увеличивает зарплату на указанный процент (число от 0 до 100);
	метод __str__, возвращающий строку «Сотрудник: {name}, зарплата: {salary}».
	В основной программе прочитай имя и зарплату через пробел. Создай объект, примени метод bonus с процентом 10, выведи строковое представление.
	Входные данные:
	Одна строка: имя и зарплата (целое число) через пробел.
	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

Сообщение об исправлении (если нужно) и строковое представление после бонуса.
Пример ввода:
Иван -500
Пример вывода:
Зарплата исправлена на 0
Сотрудник: Иван, зарплата: 0
Возможное решение:

```
class Employee:
    def __init__(self, name, salary):
        self.name = name
        if salary > 0:
            self.salary = salary
        else:
            print('Зарплата исправлена на 0')
            self.salary = 0
    def bonus(self, percent):
        self.salary = int(self.salary * (1 + percent / 100))
    def __str__(self):
        return f'Сотрудник: {self.name}, зарплата: {self.salary}'

name, salary = input().split()
emp = Employee(name, int(salary))
emp.bonus(10)
print(emp)
```

Задание 3 (написание кода)
Создай класс BankAccount с приватными атрибутами __balance (баланс) и __owner (владелец). Реализуй:
конструктор, принимающий владельца и начальный баланс (по умолчанию 0);
геттеры balance и owner через @property;
сеттер balance, который не позволяет установить отрицательное значение (если пытаются – выводит «Баланс не может быть отрицательным» и не меняет);
методы deposit(amount) и withdraw(amount) для пополнения и снятия (с проверкой, что сумма положительна и не превышает баланс);
метод __str__, возвращающий «Владелец: {owner}, баланс: {balance}».
В основной программе прочитай владельца и начальный баланс через пробел. Затем прочитай сумму для пополнения и сумму для снятия (два целых числа). Выполни операции и выведи строковое представление счёта.
Входные данные:
Первая строка – владелец и баланс; вторая строка – сумма пополнения; третья – сумма снятия.
Выходные данные:
Строковое представление счёта после операций.
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 1000
500
200
Пример вывода:
Владелец: Анна, баланс: 1300
Возможное решение:
class BankAccount:
    def __init__(self, owner, balance=0):
        self.owner = owner
        self.__balance = balance
    @property
    def balance(self):
        return self.__balance
    @balance.setter
    def balance(self, value):
        if value < 0:
            print('Баланс не может быть отрицательным')
        else:
            self.__balance = value
    @property
    def owner(self):
        return self.__owner
    @owner.setter
    def owner(self, value):
        self.__owner = value
    def deposit(self, amount):
        if amount > 0:
            self.__balance += amount
    def withdraw(self, amount):
        if 0 < amount <= self.__balance:
            self.__balance -= amount
    def __str__(self):
        return f'Владелец: {self.owner}, баланс: {self.balance}'

owner, balance = input().split()
acc = BankAccount(owner, int(balance))
dep = 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

```
wid = int(input())
acc.deposit(dep)
acc.withdraw(wid)
print(acc)
```

Задание 4 (написание кода)

Создай класс Converter с методами:

статический метод celsius_to_fahrenheit(c), который переводит градусы Цельсия в Фаренгейты по формуле F = C * 9/5 + 32;

метод класса from_fahrenheit(cls, f), который создаёт объект Converter с температурой в Цельсиях, полученной из Фаренгейтов (C = (F - 32) * 5/9), округлённой до 2 знаков.

В основной программе прочитай одно вещественное число – градусы Цельсия. Выведи результат статического метода. Затем прочитай одно вещественное число – градусы Фаренгейта, создай объект через метод класса и выведи его атрибут celsius (добавь его в конструкторе).

Входные данные:

Два вещественных числа: сначала Цельсии, затем Фаренгейты.

Выходные данные:

Сначала значение в Фаренгейтах (статический метод), затем значение в Цельсиях (из объекта).

Пример ввода:

```
25
77
```

Пример вывода:

```
77.0
25.0
```

Возможное решение:

```
class Converter:
    def __init__(self, celsius):
        self.celsius = round(celsius, 2)
    @staticmethod
    def celsius_to_fahrenheit(c):
        return c * 9/5 + 32
    @classmethod
    def from_fahrenheit(cls, f):
        c = (f - 32) * 5/9
        return cls(c)

c = float(input())
print(Converter.celsius_to_fahrenheit(c))
f = float(input())
obj = Converter.from_fahrenheit(f)
print(obj.celsius)
```

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

Создай класс Angle для представления угла в градусах и минутах. Конструктор принимает градусы и минуты (целые числа). Реализуй:
метод to_minutes() – возвращает общее количество минут;
метод __add__, который складывает два угла и возвращает новый объект Angle (при переполнении минут они переходят в градусы: 60 минут = 1 градус);
метод __str__, возвращающий строку в формате «{градусы}°{минуты}′» (например, «5°30′»).
В основной программе прочитай две строки, каждая содержит градусы и минуты через пробел. Сложи углы и выведи результат.
Входные данные:
Две строки, каждая с двумя целыми числами (градусы и минуты).
Выходные данные:
Одна строка – результат сложения.
Пример ввода:
30 40
15 50
Пример вывода:
46°30′
Возможное решение:

```
class Angle:
    def __init__(self, deg, min):
        self.deg = deg
        self.min = min
    def to_minutes(self):
        return self.deg * 60 + self.min
    def __add__(self, other):
        total = self.to_minutes() + other.to_minutes()
        d = total // 60
        m = total % 60
        return Angle(d, m)
    def __str__(self):
        return f'{self.deg}°{self.min}"

d1, m1 = map(int, input().split())
a1 = Angle(d1, m1)
d2, m2 = map(int, input().split())
a2 = Angle(d2, m2)
print(a1 + a2)
```

Задание 6 (написание кода)
Создай класс Fraction для работы с обыкновенными дробями. Конструктор принимает числитель и знаменатель, автоматически сокращает дробь (используй math.gcd). Реализуй методы:
__add__, __sub__, __mul__, __truediv__ – арифметические операции, возвращающие новый объект 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

__str__ – возвращает строку «{num}/{den}».
В основной программе прочитай две дроби (каждая в виде двух целых чисел через пробел). Выведи результаты сложения, вычитания, умножения и деления этих дробей (каждый на новой строке).
Входные данные:
Две строки, каждая с двумя целыми числами (числитель и знаменатель).
Выходные данные:
Четыре строки – результаты операций.
Пример ввода:
1 2
2 3
Пример вывода:
7/6
-1/6
1/3
3/4
Возможное решение:

```
import math

class Fraction:
    def __init__(self, num, den):
        g = math.gcd(num, den)
        self.num = num // g
        self.den = den // g
    def __add__(self, other):
        num = self.num * other.den + other.num * self.den
        den = self.den * other.den
        return Fraction(num, den)
    def __sub__(self, other):
        num = self.num * other.den - other.num * self.den
        den = self.den * other.den
        return Fraction(num, den)
    def __mul__(self, other):
        return Fraction(self.num * other.num, self.den * other.den)
    def __truediv__(self, other):
        return Fraction(self.num * other.den, self.den * other.num)
    def __str__(self):
        return f'{self.num}/{self.den}'
```

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

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

Задание 7 (написание кода)

Создай класс Point с координатами x, y. Реализуй:

конструктор с параметрами x, y (по умолчанию 0);

метод distance_to(other) – возвращает расстояние до другой точки (округлённое до двух знаков);

метод __add__, который возвращает новый объект Point с суммой координат;

метод __str__ – возвращает строку «Point({x}, {y})».

В основной программе прочитай четыре целых числа: x1, y1, x2, y2. Создай две точки, выведи расстояние между ними, затем сложи точки и выведи результат.

Входные данные:

Четыре целых числа (x1 y1 x2 y2) через пробел.

Выходные данные:

Сначала расстояние, затем строка суммы.

Пример ввода:

0 0 3 4

Пример вывода:

5.0

Point(3, 4)

Возможное решение:

```
import math

class Point:
    def __init__(self, x=0, y=0):
        self.x = x
        self.y = y
    def distance_to(self, other):
        return round(math.hypot(self.x - other.x, self.y - other.y), 2)
    def __add__(self, other):
        return Point(self.x + other.x, self.y + other.y)
    def __str__(self):
        return f'Point({self.x}, {self.y})'
```

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

```
x1, y1, x2, y2 = map(int, input().split())
p1 = Point(x1, y1)
p2 = Point(x2, y2)
print(p1.distance_to(p2))
print(p1 + p2)
```

Задание 8 (написание кода)

Создай базовый класс Animal с атрибутом name и методом speak(), который возвращает строку «...» (заглушку). Создай классы-наследники Dog и Cat, которые переопределяют метод speak(): для собаки – «Гав!», для кошки – «Мяу!». Также реализуй метод __str__, возвращающий «{name} ({класс})».

В основной программе прочитай два слова: тип животного («Dog» или «Cat») и имя. Создай соответствующий объект и выведи результат вызова speak() и строковое представление.

Входные данные:

Первая строка – тип (Dog или Cat), вторая – имя.

Выходные данные:

Две строки: сначала звук, затем строковое представление.

Пример ввода:

Dog
Шарик

Пример вывода:

Гав!
Шарик (Dog)

Возможное решение:

```
class Animal:
    def __init__(self, name):
        self.name = name
    def speak(self):
        return '...'
    def __str__(self):
        return f'{self.name} ({self.__class__.__name__})'

class Dog(Animal):
    def speak(self):
        return 'Гав!'

class Cat(Animal):
    def speak(self):
        return 'Мяу!'

t = input().strip()
```

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

```
name = input().strip()
if t == 'Dog':
    a = Dog(name)
else:
    a = Cat(name)
print(a.speak())
print(a)
```

Задание 9 (исправление ошибок)

Дана программа, которая должна создавать класс Book и выводить информацию о книге. Она содержит ошибки.

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

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

Найди и исправь ошибки.

Входные данные:

Нет (программа не считывает ввод).

Выходные данные:

Строка «Толстой - Война и мир».

Возможное решение:

```
class Book:
    def __init__(self, title, author):   # добавили self
        self.title = title
        self.author = author
    def info(self):                       # добавили self и двоеточие
        print(f'{self.author} - {self.title}')

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

Задание 10 (исправление ошибок)

Дана программа, которая должна вычислять площадь прямоугольника, но содержит ошибку.

```
class Rectangle:
    def __init__(self, width, 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

```
        self.width = width
        self.height = height
    def area(self):
        return width * height

r = Rectangle(5, 10)
print(r.area())
```

Найди и исправь ошибку.

Входные данные:

Нет.

Выходные данные:

50.

Возможное решение:

```
class Rectangle:
    def __init__(self, width, height):
        self.width = width
        self.height = height
    def area(self):
        return self.width * self.height   # добавили self.

r = Rectangle(5, 10)
print(r.area())
```

Задание 11 (исправление ошибок)

Дана программа, которая должна использовать наследование и метод super() для инициализации, но содержит ошибку.

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

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

s = Student('Анна', 2)
print(s.name, s.course)
```

Найди и исправь ошибку.

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

Нет.
Выходные данные:
Анна 2.
Возможное решение:

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

class Student(Person):
    def __init__(self, name, course):
        super().__init__(name)   # исправлено: вызов с круглыми скобками и передачей аргумента
        self.course = course

s = Student('Анна', 2)
print(s.name, s.course)
```

Задание 12 (исправление ошибок)
Дана программа, которая должна использовать свойство (@property) для контролируемого доступа к приватному атрибуту, но содержит ошибку.

```
class Temperature:
    def __init__(self, celsius):
        self.__celsius = celsius
    @property
    def celsius(self):
        return self.__celsius
    @celsius.setter
    def celsius(self, value):
        if value < -273:
            print('Температура ниже абсолютного нуля!')
        else:
            __celsius = value

t = Temperature(25)
t.celsius = -300
print(t.celsius)
```

Найди и исправь ошибку.
Входные данные:
Нет.
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

	Сообщение «Температура ниже абсолютного нуля!» и затем 25.
	Возможное решение:
	class Temperature:
	def __init__(self, celsius):
	self.__celsius = celsius
	@property
	def celsius(self):
	return self.__celsius
	@celsius.setter
	def celsius(self, value):
	if value < -273:
	print('Температура ниже абсолютного нуля!')
	else:
	self.__celsius = value # добавили self.
	
	t = Temperature(25)
	t.celsius = -300
	print(t.celsius)
	Задание 13 (краткий ответ)
	Вопрос: Какой параметр обязательно должен быть первым в определении метода экземпляра класса (кроме статических методов) и ссылается на сам объект? Запиши одним словом.
	Ответ: self
	Задание 14 (краткий ответ)
	Вопрос: Какой декоратор используется для создания метода класса, который принимает ссылку на класс (cls) вместо экземпляра? Запиши название декоратора.
	Ответ: classmethod
	Задание 15 (краткий ответ)
	Вопрос: Какой специальный метод (dunder-метод) вызывается при использовании функции print() для объекта и должен возвращать строковое представление, понятное человеку? Запиши название метода (включая двойные подчёркивания).
	Ответ: str

Шкала оценивания, нижнее значение	0
Шкала оценивания, верхнее значение	15
Шкала оценивания, минимальный проходной балл	9
WEB и API, 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)
	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). Напиши программу, которая находит минимальную температуру (temp_min) за весь период и выводит её. Если файл не содержит данных, выведи 0. Используй модуль json.
	Входные данные:
	Файл weather.json.
	Выходные данные:
	Одно вещественное число – минимальная температура.
	Пример структуры:
	{
	  "list": [
	  {"main": {"temp_min": 5.2, "temp_max": 7.8}},
	  {"main": {"temp_min": -1.3, "temp_max": 2.5}}
	  ]
	}
	Пример вывода:
	-1.3
	Возможное решение:
	import json
	with open('weather.json', encoding='utf-8') as f:
	  data = json.load(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

```
if not data['list']:
    print(0)
else:
    min_temp = min(entry['main']['temp_min'] for entry in data['list'])
    print(min_temp)
```

Задание 3 (написание кода)

В файле characters.csv в формате CSV есть столбцы name и height (рост в сантиметрах). Некоторые строки могут содержать 'NA' (неизвестно). Напиши программу, которая находит имя персонажа с наименьшим ростом (среди тех, у кого рост указан). Если несколько персонажей имеют одинаковый минимальный рост, выведи того, кто встречается первым в файле. Выведи имя персонажа.

Входные данные:

Файл characters.csv.

Выходные данные:

Имя персонажа (строка).

Пример ввода (файл):

```
name,height
Luke Skywalker,172
Darth Vader,202
Yoda,66
R2-D2,NA
```

Пример вывода:

```
Yoda
```

Возможное решение:

```
import csv
min_height = float('inf')
min_name = ''
with open('characters.csv', newline='', encoding='utf-8') as f:
    reader = csv.DictReader(f)
    for row in reader:
        if row['height'] != 'NA':
            h = int(row['height'])
            if h < min_height:
                min_height = h
                min_name = row['name']
print(min_name)
```

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. Слова считаются состоящими только из букв.

Входные данные:

Произвольное количество строк со словами.

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

Строки в формате буква: слова.
Пример ввода:
кот дом мост
стол лампа
Пример вывода:
д: дом
к: кот
л: лампа
м: мост
с: стол
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 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,
```

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

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

Задание 6 (написание кода)

Напиши программу, которая по введённой дате рождения (в формате ДД.ММ.ГГГГ) определяет, сколько полных лет исполнилось человеку на текущую дату. Используй модуль datetime и datetime.date.today(). Если день рождения ещё не наступил в текущем году, вычитай один год. Выведи количество полных лет.

Входные данные:

Одна строка с датой рождения.

Выходные данные:

Одно целое число – количество полных лет.

Пример ввода (если сегодня 15.06.2026):

10.06.2000

Пример вывода:

26

Возможное решение:

```
import datetime
birth = datetime.datetime.strptime(input(), '%d.%m.%Y').date()
today = datetime.date.today()
age = today.year - birth.year
if (today.month, today.day) < (birth.month, birth.day):
    age -= 1
print(age)
```

Задание 7 (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 8 (написание кода)
Напиши программу, которая с помощью регулярных выражений заменяет в тексте все слова «кот» (в любом регистре) на слово «пёс» (в том же регистре). Используй re.sub() с флагом re.IGNORECASE. Выведи полученную строку.
Входные данные:
Одна строка с текстом.
Выходные данные:
Строка с заменами.
Пример ввода:
Кот и котёнок. КОТ на крыше.
Пример вывода:
пёс и котёнок. пёс на крыше.
Примечание: Слово «котёнок» не должно заменяться, так как это не отдельное слово. Используй границы слова \b.
Возможное решение:

```
import re
text = input()
result = re.sub(r'\bкот\b', 'пёс', text, flags=re.IGNORECASE)
print(result)
```

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:

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('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 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.

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 11 (исправление ошибок)

Программа должна находить 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.
```

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:
	import re
	text = input()
	emails = re.findall(r'[\w.-]+@[\w.-]+', text) # исправлено
	print(' '.join(emails))
	Задание 12 (исправление ошибок)
	Программа должна принимать два числа через argparse и выводить их сумму. Код содержит ошибку – аргументы не преобразуются в числа. Исправь.
	import argparse
	parser = argparse.ArgumentParser()
	parser.add_argument('a')
	parser.add_argument('b')
	args = parser.parse_args()
	print(args.a + args.b)
	Входные данные:
	Аргументы командной строки.
	Выходные данные:
	Сумма чисел.
	Пример вызова:
	python prog.py 5 7 → вывод 12.
	Возможное решение:
	import argparse
	parser = argparse.ArgumentParser()
	parser.add_argument('a', type=int) # добавлен type=int
	parser.add_argument('b', type=int) # добавлен type=int
	args = parser.parse_args()
	print(args.a + args.b)
	Задание 13 (краткий ответ)
	Какой метод модуля json используется для записи объекта Python в файл в формате JSON? Запиши название метода.
	Ответ: dump
	Задание 14 (краткий ответ)
	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)
	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
Базы данных, 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.
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 окно с заголовком «Мой калькулятор». Размести в нём два поля для ввода целых чисел (QLineEdit) и кнопку «Сумма». При нажатии на кнопку программа должна вычислять сумму чисел из полей и выводить результат в третьем поле (QLineEdit), которое должно быть только для чтения. Используй менеджер макетов QVBoxLayout.
	Входные данные:
	Два целых числа, вводимые пользователем в поля.
	Выходные данные:
	В поле результата выводится их сумма.
	Пример:
	При вводе 5 и 8 в соответствующие поля и нажатии кнопки в поле результата появляется 13.
	Возможное решение:
	import sys
	from PyQt6.QtWidgets import QApplication, QWidget, QVBoxLayout, QLineEdit, QPushButton
	
	class Window(QWidget):
	  def __init__(self):
	  super().__init__()
	  self.setWindowTitle('Мой калькулятор')
	  layout = QVBoxLayout()
	  self.edit1 = QLineEdit()
	  self.edit2 = QLineEdit()
	  self.btn = QPushButton('Сумма')
	  self.result = QLineEdit()
	  self.result.setReadOnly(True)
	  layout.addWidget(self.edit1)
	  layout.addWidget(self.edit2)
	  layout.addWidget(self.btn)
	  layout.addWidget(self.result)
	  self.setLayout(layout)
	  self.btn.clicked.connect(self.calc)
	  def calc(self):
	  a = int(self.edit1.text())
	  b = int(self.edit2.text())
	  self.result.setText(str(a + b))
	
	if __name__ == '__main__':
	  app = QApplication(sys.argv)
	  w = Window()
	  w.show()

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

```
    sys.exit(app.exec())
```

Задание 2 (написание кода)

Создай окно с выпадающим списком (QComboBox), содержащим названия дней недели («Понедельник», «Вторник», …). При выборе любого дня в метке (QLabel) должно сразу отображаться сообщение «Выбран: <день>». Используй QVBoxLayout.

Входные данные:

Выбор пункта в списке.

Выходные данные:

Текст в метке обновляется автоматически.

Пример:

При выборе «Среда» метка показывает «Выбран: Среда».

Возможное решение:

```
import sys
from PyQt6.QtWidgets import QApplication, QWidget, QVBoxLayout, QComboBox, QLabel

class Window(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle('Выбор дня')
        layout = QVBoxLayout()
        self.combo = QComboBox()
        self.combo.addItems(['Понедельник','Вторник','Среда','Четверг','Пятница','Суббота','Воскресенье'])
        self.label = QLabel('Выберите день')
        self.combo.currentTextChanged.connect(self.label.setText)
        layout.addWidget(self.combo)
        layout.addWidget(self.label)
        self.setLayout(layout)

if __name__ == '__main__':
    app = QApplication(sys.argv)
    w = Window()
    w.show()
    sys.exit(app.exec())
```

Задание 3 (написание кода)

Напиши программу с главным окном (QMainWindow). Добавь меню «Файл» с пунктом «Выход», при выборе которого окно закрывается. Также добавь центральный виджет – метку с текстом «Главное окно». Используй QMainWindow.

Входные данные:

Нет.

Выходные данные:

При выборе пункта меню окно закрывается.

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:

```
import sys
from PyQt6.QtWidgets import QApplication, QMainWindow, QLabel

class Window(QMainWindow):
    def __init__(self):
        super().__init__()
        self.setWindowTitle('Главное окно')
        self.setCentralWidget(QLabel('Главное окно'))
        menu = self.menuBar()
        file_menu = menu.addMenu('Файл')
        file_menu.addAction('Выход', self.close)

if __name__ == '__main__':
    app = QApplication(sys.argv)
    w = Window()
    w.show()
    sys.exit(app.exec())
```

Задание 4 (написание кода)

Напиши программу, которая при нажатии на кнопку показывает информационное диалоговое окно QMessageBox с текстом «Привет, мир!» и кнопкой OK. Окно должно иметь заголовок «Сообщение».

Входные данные:

Нажатие кнопки.

Выходные данные:

Появление диалогового окна.

Возможное решение:

```
import sys
from PyQt6.QtWidgets import QApplication, QWidget, QPushButton, QMessageBox

class Window(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle('Пример')
        btn = QPushButton('Показать сообщение', self)
        btn.clicked.connect(lambda: QMessageBox.information(self, 'Сообщение', 'Привет, мир!'))

if __name__ == '__main__':
    app = QApplication(sys.argv)
```

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

```
    w = Window()
    w.show()
    sys.exit(app.exec())
```

Задание 5 (написание кода)
Создай базу данных SQLite с таблицей books (поля: id INTEGER PRIMARY KEY, title TEXT, author TEXT, year INTEGER). Напиши SQL-запрос, который выводит все книги, выпущенные после 2000 года. Запрос должен быть написан в виде строки для выполнения через sqlite3.
Входные данные:
Таблица books существует и содержит данные.
Выходные данные:
Результат запроса – список книг.
Пример:
При наличии книг с годами 1999, 2005, 2010 запрос вернёт книги 2005 и 2010.
Возможное решение:
sql
SELECT * FROM books WHERE year > 2000;
Задание 6 (написание кода)
Напиши SQL-запрос для добавления новой записи в таблицу books (название «Война и мир», автор «Лев Толстой», год 1869). Предполагается, что таблица уже создана.
Входные данные:
Данные для вставки.
Выходные данные:
В таблицу добавляется новая строка.
Возможное решение:
INSERT INTO books(title, author, year) VALUES ('Война и мир', 'Лев Толстой', 1869);
Задание 7 (написание кода)
Напиши программу на Python с использованием sqlite3, которая подключается к базе данных library.db, создаёт таблицу readers (поля: id INTEGER PRIMARY KEY, name TEXT, card_number TEXT) и добавляет одного читателя с именем «Иван Петров» и номером карты «12345». После выполнения программа должна закрыть соединение.
Входные данные:
Нет.
Выходные данные:
В базе данных появляется таблица и запись.
Возможное решение:

```
import sqlite3
con = sqlite3.connect('library.db')
con.execute('''CREATE TABLE IF NOT EXISTS readers (id INTEGER PRIMARY KEY, name TEXT, card_number TEXT)''')
con.execute('INSERT INTO readers(name, card_number) VALUES (?, ?)', ('Иван Петров', '12345'))
con.commit()
con.close()
```

Задание 8 (написание кода)
Напиши программу на PyQt, которая содержит QLineEdit для ввода текста и QLabel. При изменении текста в поле ввода метка должна автоматически отображать тот же текст (дублировать). Используй сигнал textChanged.

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:
Ввод текста пользователем.
Выходные данные:
Метка обновляется синхронно.
Пример:
При вводе «Hello» в поле, метка показывает «Hello».
Возможное решение:

```
import sys
from PyQt6.QtWidgets import QApplication, QWidget, QVBoxLayout, QLineEdit, QLabel

class Window(QWidget):
    def __init__(self):
        super().__init__()
        layout = QVBoxLayout()
        self.edit = QLineEdit()
        self.label = QLabel()
        self.edit.textChanged.connect(self.label.setText)
        layout.addWidget(self.edit)
        layout.addWidget(self.label)
        self.setLayout(layout)

if __name__ == '__main__':
    app = QApplication(sys.argv)
    w = Window()
    w.show()
    sys.exit(app.exec())
```

Задание 9 (исправление ошибок)
Дана программа, которая должна создавать окно с кнопкой, при нажатии на которую выводится сообщение «Нажато!». Программа содержит ошибку.

```
import sys
from PyQt6.QtWidgets import QApplication, QWidget, QPushButton

class Window(QWidget):
    def __init__(self):
        super().__init__()
        btn = QPushButton('Нажми меня', self)
        btn.clicked.connect(show_message)
    def show_message(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('Нажато!')

if __name__ == '__main__':
    app = QApplication(sys.argv)
    w = Window()
    w.show()
    sys.exit(app.exec())
```

Найди и исправь ошибку.

Возможное решение:

```
import sys
from PyQt6.QtWidgets import QApplication, QWidget, QPushButton

class Window(QWidget):
    def __init__(self):
        super().__init__()
        btn = QPushButton('Нажми меня', self)
        btn.clicked.connect(self.show_message)  # исправлено: добавили self.
    def show_message(self):
        print('Нажато!')

if __name__ == '__main__':
    app = QApplication(sys.argv)
    w = Window()
    w.show()
    sys.exit(app.exec())
```

Задание 10 (исправление ошибок)

Дана программа, которая должна выполнять SQL-запрос на выборку всех записей из таблицы movies и выводить их. Но она содержит ошибку.

```
import sqlite3
con = sqlite3.connect('movies.db')
result = con.execute('SELECT ALL FROM movies')
for row in result:
    print(row)
con.close()
```

Найди и исправь ошибку.

Возможное решение:

```
import sqlite3
con = sqlite3.connect('movies.db')
```

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

```
result = con.execute('SELECT * FROM movies')  # исправлено: ALL заменён на *
for row in result:
    print(row)
con.close()
```

Задание 11 (исправление ошибок)

Дана программа на PyQt, которая должна содержать вертикальный макет с двумя кнопками. Однако кнопки отображаются некорректно (перекрываются). Найди ошибку.

```
import sys
from PyQt6.QtWidgets import QApplication, QWidget, QVBoxLayout, QPushButton

class Window(QWidget):
    def __init__(self):
        super().__init__()
        layout = QVBoxLayout()
        btn1 = QPushButton('Кнопка 1')
        btn2 = QPushButton('Кнопка 2')
        layout.addWidget(btn1)
        layout.addWidget(btn2)
        # забыли установить макет
        self.setWindowTitle('Ошибка')

if __name__ == '__main__':
    app = QApplication(sys.argv)
    w = Window()
    w.show()
    sys.exit(app.exec())
```

Найди и исправь ошибку.

Возможное решение:

```
import sys
from PyQt6.QtWidgets import QApplication, QWidget, QVBoxLayout, QPushButton

class Window(QWidget):
    def __init__(self):
        super().__init__()
        layout = QVBoxLayout()
        btn1 = QPushButton('Кнопка 1')
        btn2 = QPushButton('Кнопка 2')
        layout.addWidget(btn1)
```

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

	layout.addWidget(btn2)
	self.setLayout(layout) # добавили установку макета
	self.setWindowTitle('Ошибка')
	
	if __name__ == '__main__':
	app = QApplication(sys.argv)
	w = Window()
	w.show()
	sys.exit(app.exec())
	Задание 12 (исправление ошибок)
	Дана программа, которая обновляет год выпуска фильма в таблице movies, но изменения не сохраняются. Найди ошибку.
	import sqlite3
	con = sqlite3.connect('movies.db')
	con.execute('UPDATE movies SET year = 2024 WHERE title = "Терминатор"')
	con.close()
	Возможное решение:
	import sqlite3
	con = sqlite3.connect('movies.db')
	con.execute('UPDATE movies SET year = 2024 WHERE title = "Терминатор"')
	con.commit() # добавили фиксацию изменений
	con.close()
	Задание 13 (краткий ответ)
	Вопрос: Какой метод объекта Connection в sqlite3 используется для фиксации изменений в базе данных после выполнения INSERT, UPDATE или DELETE? Запиши название метода.
	Ответ: commit
	Задание 14 (краткий ответ)
	Вопрос: Какой сигнал генерируется виджетом QPushButton при нажатии на кнопку? Запиши название сигнала.
	Ответ: clicked
	Задание 15 (краткий ответ)
	Вопрос: Какой класс в PyQt используется для создания окна, которое может содержать меню, панель инструментов и строку состояния? Запиши название класса.
	Ответ: QMainWindow

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
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 работе не менее 13 баллов (60% от максимального количества баллов). Подробное описание критериев, уровней оценивания и порядка выставления баллов представлено в Экспертном листе оценки проекта.

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Маршеня Виктория Вадимовна

серийный номер 01ba3191a94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

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.
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. 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.	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 Маршева Виктория Вадимовна
серийный номер 61ba3301c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Основные данные ДОП: ПРОДВИНУТЫЙ УРОВЕНЬ

№ п/п	Наименование поля	Значение поля
1.	Название ДОП	Python Про (продвинутый уровень, СПО)
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
7.	Кейс-задача	Название Кейс-задачи: Платформа для проведения голосований Описание проблемной ситуации: При проведении занятий, конференций, мастер-классов и других мероприятий организаторам часто требуется быстро провести опрос или викторину среди участников. Существующие сервисы не всегда подходят: часть из них имеет ограничения бесплатной версии, требует регистрации, не позволяет изменить логику работы или хранить результаты на собственном сервере. Поэтому возникла задача разработать собственную систему, которую можно использовать для проведения викторин и голосований. Цель решения: Разработать веб-приложение для проведения онлайн-викторин и голосований. Система должна позволять создавать викторины или опросы, подключать участников по ссылке или коду, принимать ответы, автоматически подсчитывать результаты и отображать их пользователям. Команда самостоятельно определяет набор возможностей системы, интерфейс и способы защиты от повторного участия. Контекст: Интерактивные платформы широко используются в образовательных организациях, компаниях и на различных мероприятиях для проведения опросов, тестов и викторин. Такие сервисы позволяют организатору создавать задания, управлять проведением мероприятия, отслеживать результаты участников в режиме реального времени и формировать итоговую статистику. В современных системах также применяются различные способы ограничения повторного участия, авторизация пользователей, использование кодов подключения и другие механизмы, обеспечивающие корректность проведения викторины или голосования. 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

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

		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»).

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, веб-разработки и работы с данными, развитие навыков проектной деятельности посредством выполнения индивидуальных и командных проектов, формирование практического опыта разработки прикладных программных решений и полноценных программных продуктов, 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 профессиональных цифровых компетенций, навыков ответственного использования ИИ-технологий и подготовка к профессиональной деятельности 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

2.	Цель и задачи ДОП	Цель программы: Формирование у обучающихся по программам среднего профессионального образования углубленных знаний и практических навыков программирования на языке Python, совершенствование алгоритмического мышления, развитие профессиональных цифровых компетенций, навыков ответственного использования ИИ-технологий и подготовка к профессиональной деятельности 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; сформировать навыки разработки графических интерфейсов с использованием библиотеки PyQt, а также работы с базами данных, включая основы языка SQL, выполнение запросов и взаимодействие с базами данных средствами Python; сформировать навыки разработки веб-приложений с использованием микрофреймворка Flask, включая создание маршрутов, использование шаблонов Jinja, обработку форм, работу с базами данных с помощью SQLAlchemy и применение принципов ORM; 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 подготовке к профессиональной деятельности 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 программного обеспечения; о современных технологиях разработки программного обеспечения; о принципах разработки графических интерфейсов, веб-приложений и работы с базами данных; о возможностях использования технологий искусственного интеллекта в процессе разработки программ; о принципах ответственного и безопасного использования информационных технологий; о возможностях профессиональной деятельности 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; основы разработки графических интерфейсов с использованием библиотеки PyQt; основы языка SQL, принципы выполнения запросов и взаимодействия с базами данных средствами Python; основы разработки веб-приложений с использованием микрофреймворка Flask, шаблонов Jinja, SQLAlchemy и принципов ORM; 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; принципы разработки графических интерфейсов, веб-приложений и взаимодействия с базами 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; разрабатывать графические интерфейсы с использованием библиотеки PyQt; работать с базами данных, использовать язык SQL, выполнять запросы и взаимодействовать с базами данных средствами Python; разрабатывать веб-приложения с использованием микрофреймворка Flask, создавать маршруты, использовать шаблоны Jinja, обрабатывать формы, работать с базами данных с помощью SQLAlchemy и применять принципы ORM; 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 разработки графических интерфейсов, веб-приложений и работы с базами данных; навыками поиска, анализа и исправления ошибок в программном коде; навыками ответственного

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

		использования технологий искусственного интеллекта; навыками проектной, исследовательской и командной деятельности при выполнении индивидуальных и командных проектов. Формируемые компетенции Программа ориентирована на формирование следующих компетенций в соответствии с ФГОС СПО по специальностям 09.02.07 Информационные системы и программирование и 09.02.09 Веб-разработка. Общие компетенции ФГОС СПО 09.02.07 Информационные системы и программирование ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам. ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности. ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие. ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами. ОК 09. Использовать информационные технологии в профессиональной деятельности. ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языках. ФГОС СПО 09.02.09 Веб-разработка ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам. ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности. ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях. ОК 04. Эффективно взаимодействовать и работать в коллективе и команде. ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках. Профессиональные компетенции ФГОС СПО 09.02.07 Информационные системы и программирование ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием. ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием. ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств. ПК 1.4. Выполнять тестирование программных модулей. ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода. ПК 9.2. Разрабатывать веб-приложение в соответствии с техническим заданием. ПК 9.3. Разрабатывать интерфейс пользователя веб-приложений в соответствии с техническим заданием. ПК 11.4. Реализовывать базу данных в конкретной системе управления базами данных. ФГОС СПО 09.02.09 Веб-разработка ПК 1.1. Проектировать информационные ресурсы. ПК 1.2. Разрабатывать интерфейсы пользователя. ПК 1.3. Интегрировать программный код в соответствующую инфраструктуру. ПК 1.4. Использовать систему контроля версий в процессе коллективной разработки. ПК 1.5. Выполнять процедуры тестирования программного кода. ПК 3.2. Создавать программный код на стороне сервера в соответствии с техническим заданием с использованием языков программирования, библиотек
		отек и фреймворков. ПК 3.3. Осуществлять отладку программного кода на стороне сервера на уровне программных модулей, межмодульных взаимодействий и взаимодействий с окружением.
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	15	1	9	4	1
2.		Использование ИИ-инструментов	2	0	1	1	0
3.		Объектно-ориентированное программирование (ООП)	5	1	3	1	0
4.		Парадигмы ООП	7	1	3	2	1
5.		Командный проект	6	0	4	2	0
6.		Промежуточная аттестация	1	0	0	0	1
7.	Модуль 2 Базы данных	Алгоритмы и структуры данных	9	1	4	3	1
8.		Графический интерфейс (Graphical User Interface, GUI)	14	1	8	4	1
9.		Базы данных	10	0	6	3	1
10.		Командный проект	2	0	2	0	0
11.		Промежуточная аттестация	1	0	0	0	1
12.	Модуль 3 WEB и API	Работа с файлами	7	1	3	2	1
13.		Ответственное использование ИИ-инструментов	2	0	1	1	0
14.		Использование библиотек	14	1	8	4	1
15.		WEB и API	6	0	4	1	1
16.		Проект	6	0	4	2	0
17.		Промежуточная аттестация	1	0	0	0	1
18.	Модуль 4 Фреймворк Flask	Модуль Flask	11	1	6	3	1
19.		SQLAlchemy	7	1	3	2	1
20.		Проект	3	0	2	1	0
21.		Структура проекта	8	1	5	2	1
22.		Кейс-задача	6	0	4	2	0
23.		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

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

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

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 и расширенных навыков программирования на языке Python, развитие алгоритмического мышления и освоение основных парадигм программирования. В рамках модуля обучающиеся изучают коллекции данных (списки, словари, множества, кортежи), функции (def, параметры, return), области видимости переменных, декораторы, работу с модулями и пакетами (import, стандартные модули math, random), основы объектно-ориентированного программирования (классы, объекты, методы классов, специальные методы __init__, __str__, __len__ и другие), принципы наследования, полиморфизма и инкапсуляции, обработку исключений (try, except), а также получают первый опыт использования технологий искусственного интеллекта в программировании и выполняют командный проект «Игра Жизнь». Задачи модуля сформировать навыки работы с коллекциями данных (списки, словари, множества, кортежи); сформировать навыки создания и использования функций (def, параметры, return); познакомить с областями видимости переменных; познакомить с использованием декораторов; познакомить с работой с модулями и пакетами (import, math, random); познакомить с основами объектно-ориентированного программирования; сформировать навыки использования специальных методов (__init__, __str__, __len__ и других); познакомить с принципами наследования, полиморфизма и инкапсуляции; познакомить с обработкой исключений (try, except); познакомить с возможностями использования технологий искусственного интеллекта при разработке программ; развить навыки командной проектной деятельности при выполнении командного проекта.	Основы Python	Знакомство с программой курса и обязательными условиями; понятие python (высокоуровневость, интерпретируемость, динамическая типизация); установка и настройка среды (openide, idle), создание проекта и виртуального окружения; pep 8; ввод/вывод (print, input), переменные, именование (snake_case), типы данных (int, float, str, bool, nonetype) и их преобразование; арифметика, приоритет, форматирование строк (конкатенация, f-строки, sep/end); разложение чисел, встроенные функции (abs, round), модуль math; коллекции (строки, списки, кортежи, множества, словари) – индексы, срезы, методы, операции, генераторы; функции (def, параметры, аргументы, return, *args, **kwargs), lambda, map, filter, reduce, all, any, enumerate, zip; итераторы, генераторы (yield, генераторные выражения); рекурсия; области видимости (global, nonlocal, legb); замыкания, декораторы (@); модули и пакеты (init.py), способы импорта, if name == 'main', стандартные модули (math, calendar, itertools), установка сторонних (pip, requests), оформление импортов по pep 8.	Теоретическая подготовка	1
		Решение задач на ввод/вывод, преобразование типов, арифметику и форматирование; обработка строк и чисел (индексы, срезы, методы, split/join, модуль math); работа со списками, кортежами, множествами, словарями и их методами, генераторы коллекций; разработка функций с параметрами, return, *args/**kwargs; применение lambda, map, filter, reduce, all, any, enumerate, zip; создание итераторов и генераторов; реализация рекурсивных алгоритмов; использование global, nonlocal, замыканий, декораторов; импорт и использование стандартных (math, calendar, itertools) и сторонних (requests) модулей; создание собственных модулей.	Практическая работа	9
		Выполнение заданий по всем темам для закрепления; решение задач на использование изученных конструкций (типы, коллекции, функции, итераторы, генераторы, рекурсия, декораторы, модули); анализ и отладка кода, поиск и исправление ошибок; тестовые задания на понимание теории (сопоставление, выбор ответа, определение результата); изучение дополнительных материалов, подготовка к текущему контролю.	Самостоятельная работа	4
		Вопросы на знание видов ИИ-инструментов, их возможностей и ограничений, правил безопасного и осознанного использования. В практическую часть входят задания на анализ кода, сгенерированного ИИ, поиск и исправление ошибок, а также написание программ с использованием ИИ для объяснения новых тем и последующей самостоятельной реализации.	Текущий контроль	1
	Использование ИИ-инструментов		Теоретическая подготовка	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

Планируемые результаты обучения По итогам освоения модуля обучающийся будет: знать: основы работы с коллекциями данных (списки, словари, множества, кортежи); основы работы с функциями (def, параметры, return); области видимости переменных; назначение декораторов; основы работы с модулями и пакетами (import, math, random); основы объектно-ориентированного программирования; назначение специальных методов (__init__, __str__, __len__ и других); принципы наследования, полиморфизма и инкапсуляции; основы обработки исключений (try, except); назначение технологий искусственного интеллекта в программировании. уметь: работать с коллекциями данных; создавать и использовать функции; использовать области видимости переменных; применять декораторы в простых задачах; использовать модули и пакеты Python; создавать классы и объекты; использовать специальные методы (__init__, __str__, __len__ и других); применять наследование, полиморфизм и инкапсуляцию; обрабатывать исключения (try, except); использовать технологии искусственного интеллекта для анализа и улучшения программного кода; выполнять командный проект. владеть: навыками работы с коллекциями данных; навыками использования функций; навыками применения декораторов; навыками работы с модулями и пакетами; навыками объектно-ориентированного программирования; навыками использования специальных методов; навыками обработки исключений; навыками применения принципов ООП; навыками командной проектной деятельности.		Анализ и исправление кода, сгенерированного ИИ (задача на согласование слова «копейка» с числом), написание программы с осознанным применением ИИ-инструмента для получения объяснения правил определения високосного года и тестов, а затем самостоятельное составление решения, а также использование ИИ для знакомства с новой конструкцией (цикл while) и последующее дописывание кода для вычисления суммы цифр числа. Все задания выполняются с акцентом на проверку ответов ИИ и самостоятельное осмысление.	Практическая работа	1
		Закрепление навыков использования ИИ-инструментов: самостоятельное применение ИИ для проверки своих решений, поиска ошибок и альтернативных подходов, но без копирования готовых ответов; фиксация собственных выводов о работе ИИ; изучение дополнительных материалов по возможностям и ограничениям ИИ в программировании; подготовка к текущему контролю.	Самостоятельная работа	1
			Текущий контроль	0
	Объектно-ориентированное программирование (ООП)	Знакомство с понятием специальных (магических) методов Python, их назначением и автоматическим вызовом при выполнении операций с объектами; изучается метод инициализации __init__ для настройки экземпляров при создании, методы строкового представления __str__ (для пользователей) и __repr__ (для отладки). Рассматривается перегрузка арифметических операторов через методы __add__, __sub__, __mul__, __truediv__, а также расширенные присваивания (__iadd__) и отражённые методы (__radd__). Изучаются методы сравнения __eq__, __ne__, __lt__, __gt__, __le__, __ge__ и использование декоратора @total_ordering для автоматической генерации недостающих сравнений. Также даётся обзор других специальных методов: __len__, __getitem__, __setitem__, __delitem__, __contains__, __call__, __abs__, __bool__.	Теоретическая подготовка	1
		Разработка классов с реализацией специальных методов: инициализации, строкового представления (__str__), арифметических операций (сложение, вычитание, умножение, деление) для класса обыкновенных дробей, методов сравнения для дробей и времени, а также создание класса последовательности с перегрузкой сложения для объединения уникальных элементов. Отрабатываются навыки использования @total_ordering, работы с атрибутами через __dict__, применения статических методов для вспомогательных вычислений (сокращение дробей, поиск общего знаменателя). Выполняются задания на создание классов Sequence, Fraction, Time, Field, Check, Vector, Brick с перегрузкой операторов и корректным строковым выводом.	Практическая работа	3
		Выполнение тестовых заданий на знание специальных методов и их соответствия операторам, а также практических задач: создание класса Sequence с перегрузкой сложения и класса Time с методами сравнения. Учащиеся закрепляют понимание различий между __str__ и __repr__, назначения __iadd__ и __add__, условий вызова __bool__, применения декоратора @total_ordering и работы с функциями isinstance, getattr, hasattr, setattr, delattr.	Самостоятельная работа	1
			Текущий контроль	0
	Парадигмы ООП	Знакомство с понятием специальных (магических) методов Python, их назначением и автоматическим вызовом при выполнении операций с объектами; изучается метод инициализации __init__ для настройки экземпляров при создании, методы строкового представления __str__ (для пользователей) и __repr__ (для отладки). Рассматривается перегрузка арифметических операторов через методы __add__, __sub__, __mul__, __truediv__, а также расширенные присваивания (__iadd__) и отражённые методы (__radd__). Изучаются методы сравнения __eq__, __ne__, __lt__, __gt__, __le__, __ge__ и использование декоратора @total_ordering для автоматической генерации недостающих сравнений. Также даётся обзор других специальных методов: __len__, __getitem__, __setitem__, __delitem__, __contains__, __call__, __abs__, __bool__.	Теоретическая подготовка	1

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Маршенко Виктория Вадимовна
серийный номер 01ba3191894f2eb04e6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

		Создание классов с наследованием: разработка классов Person и Student с переопределением методов __init__ и get_name, использование super() для вызова родительского конструктора; изучение множественного наследования и порядка mro через создание иерархии классов и вывод __mro__; реализация полиморфизма через создание иерархии геометрических фигур (прямоугольник, окружность, треугольник, прямоугольный треугольник) с едиными методами get_area() и get_perimeter(); применение инкапсуляции при создании класса Point с сокрытием атрибутов и использованием @property для управляемого доступа с валидацией значений; разработка класса Password с проверкой длины и наличия специальных символов через собственные исключения; создание программы-калькулятора с обработкой исключений и пользовательским исключением OperationError; перехват и обработка различных типов исключений (valueerror, zerodivisionerror) с выводом соответствующих сообщений.	Практическая работа	3
		Выполнение тестовых заданий на знание принципов наследования, порядка mro, работы super(), видов полиморфизма, механизмов инкапсуляции и обработки исключений; решение практических задач на создание иерархий классов с наследованием, переопределением методов и использованием полиморфизма; разработку классов с инкапсуляцией, геттерами и сеттерами через @property; написание программ с обработкой исключений и созданием собственных классов исключений; анализ и отладку кода с наследованием и исключениями, поиск и исправление ошибок.	Самостоятельная работа	2
		Решение задач с использованием lambda и функций высшего порядка filter() и map() для отбора и преобразования данных. Создание класса Book с конструктором и методом info; класса Student с валидацией курса, статическим методом и методом класса from_string; класса Rectangle с приватными атрибутами, свойствами и сеттерами с проверкой; иерархии Employee и Manager с переопределением метода get_info; класса Fraction с сокращением, перегрузкой __add__, __eq__, __lt__; а также задач на применение @property, наследования, специальных методов и обработки исключений (например, калькулятор с собственным исключением, класс Password с проверкой длины и спецсимволов).	Текущий контроль	1
	Командный проект		Теоретическая подготовка	0
		Разработка логики игры «Жизнь» в составе команды: создание классов Cell и Field с реализацией всех требуемых методов (get_neighbours_count, get_next_field, next_generation, set_figure), определение структуры для хранения фигур в словаре FIGURES с использованием лямбда-функций, возвращающих координаты живых клеток относительно заданной позиции. Выполняется установка библиотеки pygame через pip, создание модуля life.py с логикой и main.py с готовым графическим интерфейсом, интеграция разработанной логики с интерфейсом, тестирование работы программы с различными начальными условиями (случайное заполнение, размещение фигур r-pentamino и planner в центре или по координатам). На этапе защиты проекта проводится презентация результата: демонстрация работающей программы, объяснение ключевых алгоритмических решений, анализ сильных сторон и возможных улучшений, описание распределения задач и вклада каждого участника при командной работе, а также обсуждение возникших трудностей и способов их преодоления.	Практическая работа	4
		Доработка проекта: добавление информации о дополнительных фигурах в константу FIGURES, улучшение пользовательского интерфейса (если требуется), проведение дополнительных тестов для проверки корректности работы с различными фигурами и размерами поля. Учащиеся готовятся к защите проекта, оформляя описание проекта, его назначение, основные этапы разработки и полученные результаты, фиксируя возникшие сложности и способы их решения. Также рекомендуется самостоятельно изучить дополнительные материалы по клеточным автоматам и возможным расширениям игры (например, изменение правил, добавление цветовой индикации).	Самостоятельная работа	2
			Текущий контроль	0

Контур Крипто

Документ подписан квалифицированной электронной подписью 04.08.2026

владелец Марцева Виктория Вадимовна

серийный номер 01c319b946b1e4f607a019ab1

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 2 Базы данных Модуль направлен на формирование у обучающихся практических навыков работы со структурами данных, алгоритмами, графическими интерфейсами и базами данных в языке Python. В рамках модуля обучающиеся изучают структуры данных (списки, стек, дек, очередь), алгоритмы поиска и сортировки (линейный и двоичный поиск, сортировки выбором, пузырьком и вставками, быструю сортировку), оценку сложности алгоритмов (О-нотация), основы разработки графических интерфейсов с использованием PyQt (виджеты: QWidget, QLabel, QPushButton, QLineEdit, QTableWidget и другие), работу с Qt Designer, обработку событий (signals/slots), создание окон и диалогов, основы баз данных и языка SQL, выполнение SQL-запросов (SELECT, INSERT, UPDATE, DELETE и другие), работу с базами данных в Python (sqlite3), а также выполняют командный проект «Список задач». Задачи модуля сформировать навыки работы со структурами данных в Python (списки, стек, дек, очередь); познакомить с алгоритмами поиска и сортировки (линейный и бинарный поиск, сортировки выбором, пузырьком, вставками); сформировать представление о сложности алгоритмов (О-нотация); развить навыки решения алгоритмических задач; сформировать навыки разработки графических интерфейсов с использованием библиотеки PyQt; познакомить с основными виджетами (QWidget, QLabel, QPushButton, QLineEdit, QTableWidget и другие); сформировать навыки использования Qt Designer для проектирования интерфейсов; познакомить с обработкой событий (signals/slots); сформировать навыки создания окон и диалогов в приложениях; сформировать навыки работы с базами данных; познакомить с основами SQL (SELECT, INSERT, UPDATE, DELETE и другие); сформировать навыки выполнения SQL-запросов; познакомить с работой с базами данных в Python (sqlite3); развить навыки разработки прикладных приложений с графическим интерфейсом; развить навыки проектной деятельности при выполнении командного проекта; Планируемые результаты обучения По итогам освоения модуля обучающийся будет:	Алгоритмы и структуры данных	Изучение структур данных как способов организации и хранения информации: статические и динамические массивы (принцип релокации, реализация через списки Python с методами append, insert, pop, del), связные списки (односвязный список, узел Node, ссылки next, операции вставки в начало, конец и произвольную позицию), стек (LIFO, методы push, pop, size, реализация на списке), очередь (FIFO, push, pop, size) и дек (двусторонняя очередь, push_front, push_back, pop_front, pop_back), а также хеш-таблицы как основа множеств и словарей (принцип хеширования, коллизии). Рассматриваются алгоритмы сортировки: сортировка пузырьком (попарное сравнение и обмен, оптимизация с флагом), сортировка выбором (поиск максимума и обмен с последней позицией), сортировка вставками (поочерёдное включение элементов в отсортированную часть), сортировка слиянием (разделяй и властвуй, рекурсивное разбиение и слияние двух отсортированных массивов) и быстрая сортировка (выбор опорного элемента, разделение на левую, центральную и правую части, рекурсия). Изучаются алгоритмы поиска: линейный поиск (последовательный перебор, O(n)) и двоичный поиск (только на отсортированном массиве, деление пополам, O(log n)), а также левый и правый двоичный поиск для обработки повторяющихся элементов. Вводится понятие сложности алгоритмов (временная и пространственная), асимптотическая оценка с использованием О-нотации (Big O): константная O(1), линейная O(n), логарифмическая O(log n), линейно-логарифмическая O(n log n), квадратичная O(n²). Анализируется сложность различных операций для динамических массивов, связных списков и хеш-таблиц, а также сравниваются алгоритмы сортировки по времени в лучшем, среднем и худшем случаях и используемой памяти.	Теоретическая подготовка	1
		Реализация связного списка с поиском элемента по значению и выводом его индекса (или -1), создание стека на основе списка с методами push, pop, size, а также очереди на односвязном списке с указателями head и tail для обеспечения O(1) при добавлении и извлечении. Отрабатываются алгоритмы сортировки: сортировка выбором с выводом состояния списка после каждого прохода, сортировка слиянием и быстрая сортировка с рекурсивной реализацией. Выполняются задачи на линейный поиск и двоичный поиск (включая левый двоичный поиск для первого вхождения элемента). Реализуется эффективная очередь с методами push, pop, size с проверкой на пустоту и выводом ошибок. Все задания сопровождаются анализом временной сложности реализованных операций.	Практическая работа	4
		Выполнение заданий на реализацию структур данных (стек, очередь на связном списке) и алгоритмов сортировки и поиска с последующей оценкой их сложности, а также решение задач на применение левого и правого двоичного поиска, сортировку слиянием и быструю сортировку. Проводится анализ и сравнение эффективности различных структур данных и алгоритмов при решении конкретных задач, отработка навыков определения асимптотической сложности по коду, подготовка к текущему контролю.	Самостоятельная работа	3
		Поиск в связном списке, вычисление выражений в обратной польской записи, реализация очереди с методами, сортировка вставками с выводом состояний, правый двоичный поиск, поиск и исправление ошибок в готовом коде, тестовые задания на сопоставление терминов.	Текущий контроль	1

Контур Крипто
владелец Маршеня Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3191c94f2eb04c5f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

знать: основные структуры данных Python (списки, стек, дек, очередь); алгоритмы поиска и сортировки (линейный и бинарный поиск, сортировки выбором, пузырьком и вставками); основы сложности алгоритмов (O-нотация); основы разработки графических интерфейсов с использованием PyQt; назначение основных виджетов (QWidget, QLabel, QPushButton, QLineEdit, QTableWidget и другие); принципы работы Qt Designer; основы обработки событий (signals/slots); основы работы с окнами и диалогами; основы баз данных; основы языка SQL (SELECT, INSERT, UPDATE, DELETE и другие); основы работы с sqlite3 в Python. уметь: использовать структуры данных при решении задач; применять алгоритмы поиска и сортировки; оценивать сложность алгоритмов; создавать графические интерфейсы с использованием PyQt; использовать основные виджеты (QLabel, QPushButton, QLineEdit, QTableWidget); применять Qt Designer для проектирования интерфейсов; обрабатывать события через signals/slots; создавать окна и диалоги; выполнять SQL-запросы (SELECT, INSERT, UPDATE, DELETE и другие); работать с базами данных в Python (sqlite3); разрабатывать прикладные приложения с графическим интерфейсом; выполнять командный проект. владеть: навыками работы со структурами данных; навыками реализации алгоритмов поиска и сортировки; навыками оценки сложности алгоритмов; навыками разработки GUI-приложений; навыками работы с базами данных; навыками выполнения SQL-запросов; навыками разработки прикладных программных решений; навыками командной проектной деятельности.	Графический интерфейс (Graphical User Interface, GUI)	Изучение структур данных (массивы, связные списки, стек, очередь, дек, хеш-таблицы), алгоритмов сортировки (пузырьком, выбором, вставками, слиянием, быстрая) и поиска (линейный, двоичный, левый и правый двоичный), а также оценки сложности алгоритмов с использованием O-нотации (Big O). Рассматриваются основы создания графических интерфейсов на PyQt: виджеты (QLabel, QPushButton, QLineEdit), менеджеры макетов (QVBoxLayout, QHBoxLayout, QGridLayout, QFormLayout), абсолютное позиционирование, работа с Qt Designer (создание .ui файлов, подключение через uic или pyuic6), обработка событий с помощью сигналов и слотов, перехват событий мыши и клавиатуры, создание главного окна QMainWindow (меню, панели инструментов, док-виджеты, строка состояния) и диалоговых окон (QMessageBox, QInputDialog).	Теоретическая подготовка	1
		Реализация связного списка с поиском элемента, стека и очереди с методами push, pop, size; написание алгоритмов сортировки (выбором, слиянием, быстрая) и поиска (линейный, двоичный, левый двоичный); разработку GUI-приложений: калькулятор ИМТ на Tkinter, генератор паролей на PyQt, программа для арифметических операций над дробями с макетами, форма авторизации и регистрации, анализатор текста, конвертер валют с вкладками, текстовый редактор с меню и панелью инструментов; создание интерфейсов в Qt Designer и их подключение; обработку событий мыши и клавиатуры (калькулятор с вводом с клавиатуры); использование диалоговых окон (QMessageBox, QInputDialog).	Практическая работа	8
		Решение дополнительных задач на структуры данных (дек на двусвязном списке), оптимизацию алгоритмов сортировки, определение сложности фрагментов кода и выбор оптимальных структур; создание собственных GUI-приложений с использованием изученных виджетов и макетов; углублённое изучение документации PyQt; подготовку к текущему контролю и промежуточной аттестации.	Самостоятельная работа	4
		Реализация структуры данных (очередь, стек), алгоритмов сортировки и поиска; создание GUI-приложений с заданным функционалом с использованием Qt Designer и обработкой событий; задачи на поиск и исправление ошибок в готовом коде, определение результата выполнения фрагментов программ, тестовые задания на сопоставление терминов.	Текущий контроль	1
	Базы данных		Теоретическая подготовка	0
		Выполнение SQL-запросов в SQLiteStudio с условиями, группировкой, сортировкой и соединениями, написание Python-программ для CRUD-операций с БД через sqlite3, создание GUI-приложения на PyQt для работы с базой данных фильмов (вывод таблицы, отображение и выбор жанров, добавление, редактирование, удаление записей с подтверждением через диалоги), использование многотабличных запросов и атрибута lastrowid, включение поддержки внешних ключей.	Практическая работа	6
		Выполнение SQL-запросов в SQLiteStudio с условиями, группировкой, сортировкой и соединениями, написание Python-программ для CRUD-операций с БД через sqlite3, создание GUI-приложения на PyQt для работы с базой данных фильмов (вывод таблицы, отображение и выбор жанров, добавление, редактирование, удаление записей с подтверждением через диалоги), использование многотабличных запросов и атрибута lastrowid, включение поддержки внешних ключей.	Самостоятельная работа	3
		Написание запросов на выборку с условиями и сортировкой, добавление, обновление и удаление данных, запросы с группировкой и агрегацией, соединение таблиц, использование подзапросов, задачи на поиск и исправление ошибок в SQL-коде.	Текущий контроль	1
	Командный проект		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

		Разработка индивидуального проекта «Список задач»: создание базы данных с таблицами categories и tasks с учётом ограничений; реализация класса Categories (инициализация, загрузка категорий в QListWidget, добавление категории через QInputDialog, удаление с подтверждением QMessageBox); реализация класса Tasks (инициализация интерфейса, загрузка задач с фильтрацией по категориям и отображением статуса выполнения через чекбоксы, загрузка категорий в QComboBox, вывод подробностей задачи при выделении с обновлением поля done, добавление задачи с проверкой уникальности названия, удаление задачи с подтверждением, открытие модального окна категорий); интеграция всех компонентов, тестирование работоспособности приложения, добавление тестовых данных. Доработка проекта: добавление проверок на дублирование названий задач и категорий, обработку исключений при работе с БД, реализацию дополнительного функционала (поиск задач, сортировка, экспорт списка), улучшение пользовательского интерфейса, подготовку проекта к демонстрации и добавление в портфолио.	Практическая работа	2
			Самостоятельная работа	0
			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 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), логирование (logging), основы работы с API (requests), парсинг данных (HTML-разбор), основы работы с библиотекой vk_api, систему контроля версий Git, а также выполняют проект «VK бот» и знакомятся с принципами ответственного использования технологий искусственного интеллекта. 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;	Работа с файлами	Изучение стандартного потока ввода sys.stdin и его методов (read, readlines), работы с файлами через функцию open (режимы чтения 'r', записи 'w', добавления 'a'), использование менеджера контекста with для автоматического закрытия файлов, методов чтения (read, readlines) и записи (write, writelines); изучение форматов обмена данными JSON и CSV: структура JSON, модуль json с методами load, loads, dump, dumps и параметрами ensure_ascii, indent, sort_keys; структура CSV, модуль csv с reader, writer, DictReader, DictWriter и параметрами delimiter, quotechar, fieldnames; преобразование данных между Python-объектами и файлами форматов JSON и CSV.	Теоретическая подготовка	1
		Использование sys.stdin для ввода произвольного количества строк, чтение и запись текстовых файлов с обработкой данных (суммирование чисел, сортировка строк), чтение JSON-файлов с извлечением данных по ключам (поиск максимальной температуры), чтение CSV-файлов с использованием reader и DictReader (поиск персонажа с наибольшим ростом), запись данных в CSV с использованием writer и DictWriter, преобразование JSON-данных в CSV с сортировкой, обработка ошибок при работе с файлами и форматами данных.	Практическая работа	3
		Выполнение дополнительных заданий на чтение и запись файлов различных форматов, обработку данных из JSON и CSV с фильтрацией и сортировкой, преобразование данных между форматами, анализ и отладку кода, подготовку к текущему контролю.	Самостоятельная работа	2
		Чтение файлов и вычисление сумм, поиск экстремумов в JSON и CSV, группировка файлов по расширению через sys.stdin, преобразование форматов, поиск и исправление ошибок в коде, анализ сгенерированных ИИ решений и их доработка с учётом ограничений.	Текущий контроль	1
	Ответственное использование ИИ инструментов		Теоретическая подготовка	0

Контур.Крипто
владелец Марсиян Вадим Олегович
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

развить навыки разработки программ, взаимодействующих с внешними сервисами;
развить навыки проектной деятельности при выполнении командного проекта;

Планируемые результаты обучения
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.
принципы ответственного использования ИИ-инструментов.

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;
использовать ИИ-инструменты для анализа и оптимизации кода;
выполнять командный проект;

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 применения ИИ-инструментов в процессе разработки решений.
навыками командной проектной деятельности.

ИИ-инструментов	Чтение и запись текстовых файлов с обработкой данных (суммирование чисел, сортировка строк), чтение JSON-файлов с извлечением данных по ключам, чтение и запись CSV-файлов с использованием reader, writer, DictReader, DictWriter, преобразование данных между форматами, анализ и исправление кода, сгенерированного ИИ, поиск ошибок в готовых решениях, генерацию тестов с помощью ИИ, осознанное использование ИИ для получения объяснений и подсказок без копирования готового кода.	Практическая работа	1
	Выполнение дополнительных заданий на чтение и запись файлов различных форматов, обработку данных из JSON и CSV с фильтрацией и сортировкой, преобразование данных между форматами, анализ и отладку кода, подготовку к текущему контролю, закрепление навыков ответственного использования ИИ-инструментов с фиксацией собственных выводов о работе ИИ.	Самостоятельная работа	1
		Текущий контроль	0
Использование библиотек	Работа с файлами и форматами JSON/CSV, командную строку и argparse, datetime и schedule, регулярные выражения и re, тестирование (assert, doctest, unittest), логирование (logging), ответственное использование ИИ-инструментов.	Теоретическая подготовка	1
	Чтение и запись файлов, обработку JSON и CSV, парсинг аргументов командной строки (калькулятор, генератор чисел), работу с датой и временем (выходные дни, календарь, таймер), использование schedule, поиск и замену с регулярными выражениями (даты, телефоны, ФИО, номера), написание тестов с assert, doctest и unittest, логирование событий с разными уровнями в файл.	Практическая работа	8
	Выполнение заданий по всем темам для закрепления, решение задач на работу с файлами и форматами, командной строкой, датой и временем, регулярными выражениями, написание тестов и логирование, анализ и отладку кода, подготовку к текущему контролю.	Самостоятельная работа	4
	Чтение и запись файлов, обработка JSON и CSV, парсинг аргументов, работа с datetime, использование регулярных выражений для поиска и замены, написание тестов (assert, doctest, unittest), логирование, поиск и исправление ошибок в готовом коде, тестовые задания на сопоставление терминов.	Текущий контроль	1
WEB и API		Теоретическая подготовка	0
	Выполнение запросов к открытым API (получение вакансий с портала «Работа России» с параметрами, работа с DaData для подсказок по паспортным данным с использованием токена), написание программ для парсинга статических веб-страниц с Beautiful Soup (подсчёт количества PEP на странице, вывод версий PyQt6 и дат релизов, извлечение списка стандартных функций Python со ссылками на документацию), обработку ответов в формате JSON.	Практическая работа	4
	Выполнение заданий на работу с API (запросы с параметрами, авторизация по токену, обработка ответов) и парсинг веб-страниц (поиск элементов по тегам и атрибутам, извлечение данных), анализ и отладку кода, подготовку к текущему контролю.	Самостоятельная работа	1
	Написание программ для работы с API, парсинга веб-страниц, обработки CSV и JSON, создания утилит с аргументами командной строки, поиск и исправление ошибок в готовом коде, тестовые задания на сопоставление терминов.	Текущий контроль	1
Проект		Теоретическая подготовка	0
	Выполнение GET-запросов к открытым API (получение вакансий с портала «Работа России» с параметрами), работу с DaData для подсказок по паспортным данным с использованием токена, парсинг статических веб-страниц с Beautiful Soup (подсчёт количества PEP, вывод версий PyQt6 и дат релизов, извлечение списка стандартных функций Python со ссылками на документацию), создание VK-бота (эхо-бот и бот для проверки орфографии через Яндекс Спеллер) с использованием библиотеки vk_api и Long Poll, работу с Git через OpenIDE (клонирование удалённого репозитория, добавление файлов в отслеживаемые, выполнение commit и push, заполнение README.md с использованием Markdown, выполнение pull для получения изменений).	Практическая работа	4

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Марцинкевич Виктор Владимирович
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.06.2027

		Выполнение заданий по всем темам для закрепления, решение задач на работу с API (запросы с параметрами, авторизация по токену, обработка ответов), парсинг веб-страниц (поиск элементов по тегам и атрибутам, извлечение структурированных данных), создание собственного VK-бота с дополнительным функционалом, работу с Git (создание репозитория, коммиты, push, pull, оформление README), анализ и отладку кода, подготовку к текущему контролю.	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
Модуль 4 Фреймворк Flask Модуль направлен на формирование у обучающихся практических навыков веб-разработки на Python с использованием микрофреймворка Flask, освоение принципов создания веб-приложений, работы с шаблонами, базами данных и структурой проекта. В рамках модуля обучающиеся изучают Flask (маршруты, GET/POST), рендеринг HTML (render_template), шаблонизатор Jinja2 (переменные, условия, циклы), обработку форм (request.form), работу с базами данных через SQLAlchemy (модели, CRUD-операции), структуру веб-приложений, основы авторизации и регистрации, а также выполняют проект «Укорачиватель ссылок» и кейс-задачу «Платформа для проведения голосований». Задачи модуля сформировать навыки разработки веб-приложений с использованием Flask; познакомить с маршрутизацией (GET, POST); сформировать навыки рендеринга HTML-страниц (render_template); познакомить с Jinja2 (переменные, условия, циклы); сформировать навыки обработки форм (request.form); сформировать навыки работы с базами данных; познакомить с SQLAlchemy (модели, CRUD); сформировать навыки структурирования веб-приложений; развить навыки проектной деятельности при выполнении командного проекта; развить навыки решения прикладной кейс-задачи. Планируемые результаты обучения По итогам освоения модуля обучающийся будет: знать: основы Flask и маршрутизации;	Модуль Flask	Знакомство с микрофреймворком Flask (установка, создание приложения, режим отладки), маршрутизацией (статические и динамические URL, конвертеры, add_url_rule, url_for), основами HTML и CSS (структура, теги, стили, Bootstrap), работой с шаблонами (render_template, передача переменных), шаблонизатором Jinja2 (вывод переменных, условия, циклы с loop, фильтры, наследование и включение шаблонов), формами (Flask-WTF, классы форм, поля, валидаторы, CSRF-защита, обработка POST-запросов, перенаправление), а также статическими файлами.	Теоретическая подготовка	1
		Создание маршрутов для вычислений (умножение, сумма, простые числа, калькулятор), рендеринг шаблонов со списками и словарями, использование циклов, условий и фильтров в Jinja, подключение Bootstrap, разработку форм обратной связи и добавления новостей с валидацией и перенаправлением, интеграцию всех частей в проект «Новостной портал» (главная, детальная страница, форма добавления, базовый шаблон с меню и подвалом).	Практическая работа	6
		Выполнение заданий по созданию маршрутов с конвертерами, разработку шаблонов с наследованием и фильтрами, создание форм с различными полями и валидаторами, доработку проекта «Новостной портал» с добавлением новых функций, анализ и отладку кода, подготовку к текущему контролю.	Самостоятельная работа	3
		Создание маршрутов с вычислениями и динамическими параметрами, рендеринг шаблонов с циклами и условиями, разработка форм с валидацией и перенаправлением, использование наследования шаблонов, подключение статических файлов, обработка GET и POST-запросов, поиск и исправление ошибок в коде Flask-приложений, тестовые задания на сопоставление терминов и конструкций Jinja.	Текущий контроль	1
	SQLAlchemy	Знакомство с технологией ORM (Object-Relational Mapping) и библиотекой SQLAlchemy, подключение SQLAlchemy к Flask через flask-sqlalchemy, настройку подключения к SQLite, создание моделей (классы, наследуемые от db.Model, поля с типами Integer, String, Text, DateTime, Boolean, параметры primary_key, unique, nullable, default), определение связей между таблицами (ForeignKey, relationship, back_populates), создание таблиц через db.create_all, работу с данными через CRUD-операции (создание, чтение, обновление, удаление объектов с использованием db.session.add, db.session.commit, db.session.delete), методы Query (all, get, filter с операторами сравнения и contains, order_by, limit, offset), использование flask shell для интерактивной работы с БД.	Теоретическая подготовка	1

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 Flask и маршрутизации; основы рендеринга HTML (render_template); основы Jinja2; основы обработки форм (request.form); основы работы с базами данных; основы SQLAlchemy; структуру веб-приложений; основы авторизации и аутентификации. уметь: создавать веб-приложения на Flask; настраивать маршруты (GET, POST); рендерить HTML-страницы; использовать Jinja2; обрабатывать формы; работать с SQLAlchemy (модели, CRUD); проектировать структуру веб-приложения; выполнять командный проект; решать прикладную кейс-задачу. владеть: навыками веб-разработки на Flask; навыками работы с шаблонами и формами; навыками работы с SQLAlchemy; навыками проектирования веб-приложений; навыками реализации CRUD-логики; навыками командной разработки; навыками решения кейс-задач.		Подключение базы данных к проекту flask_news, создание модели News и Category с полями и связями, выполнение операций CRUD через flask shell, доработку шаблонов для отображения списка категорий и новостей с фильтрацией по категориям, добавление в форму выбора категории через SelectField с динамическим заполнением, создание маршрута для вывода новостей конкретной категории, отображение категории в карточке новости, добавление даты создания для новых записей.	Практическая работа	3
		Выполнение заданий по созданию моделей с различными типами полей и связями, реализацию CRUD-операций через flask shell и интерфейс приложения, доработку проекта с добавлением новых функций (редактирование и удаление новостей, управление категориями), анализ и отладку кода, подготовку к текущему контролю.	Самостоятельная работа	2
		Создание моделей с полями и связями, выполнение CRUD-операций, создание форм с валидацией, интеграция БД в Flask-приложение с отображением данных, реализация генерации уникальных кодов, обработка перенаправлений и увеличение счётчиков, поиск и исправление ошибок в коде работы с SQLAlchemy и формами, тестовые задания на сопоставление терминов.	Текущий контроль	1
	Проект		Теоретическая подготовка	0
		Создание Flask-приложения с маршрутами /, /urls и /<short>, подключение SQLite через SQLAlchemy и создание модели URLmodel с полями id, original_url, short, visits, created_at, создание формы URLForm с валидацией ссылок, реализацию функции get_short для генерации уникальных коротких ссылок, доработку шаблонов с наследованием (base.html, navigation.html, index.html, urls.html), отображение ссылок из БД на странице /urls с сортировкой от новых к старым, обработку POST-запросов для создания новых записей и перенаправление на страницу со ссылками, реализацию эндпоинта /<short> для перенаправления по оригинальной ссылке с увеличением счётчика посещений.	Практическая работа	2
		Выполнение заданий по доработке проекта «Укорачиватель ссылок» (добавление возможности удаления ссылок, редактирования, поиска, пагинации, улучшение интерфейса), создание аналогичных проектов с использованием изученных подходов, анализ и отладку кода, подготовку к текущему контролю.	Самостоятельная работа	1
			Текущий контроль	0
	Структура проекта		Теоретическая подготовка	0
		Структурирование проекта flask_news и flask_shorterer: создание пакета app, перенос static, templates, моделей, форм и представлений в отдельные файлы, настройка конфигурации через config.py и .env, установка зависимостей через requirements.txt, создание README.md, инициализация Git-репозитория и отправка на GitHub; интеграция Flask-Login в проект (создание модели User с хэшированием пароля, форм регистрации и входа, шаблонов, функций login, registration, logout, ограничение доступа к add_news и add_category через @login_required, добавление flash-сообщений); публикация проекта на PythonAnywhere (клонирование репозитория, настройка виртуального окружения, создание .env, настройка WSGI и запуск).	Практическая работа	5
		Доработка проектов с добавлением функций управления категориями и пользователями, обновление файлов зависимостей и README, самостоятельную публикацию проектов на PythonAnywhere, анализ и отладку кода, подготовку к текущему контролю.	Самостоятельная работа	2
		Создание модульной структуры проекта, модели пользователя с хэшированием форм и маршрутов для регистрации и входа, защита маршрутов через @login_required, использование flash-сообщений, настройка деплоя на PythonAnywhere, поиск и исправление ошибок в коде аутентификации и конфигурации, тестовые задания на сопоставление терминов.	Текущий контроль	1
	Кейс-задача		Теоретическая подготовка	0

Контур Крипто

владелец Мартынюк Виктория Вадимовна

Документ подписан квалифицированной электронной подписью 04.08.2026

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

		Анализ проблемной ситуации, самостоятельное определение проблемы, формулировку цели и ожидаемого результата проекта, разбиение работы на задачи и планирование с использованием Kanban-доски, разработку решения в команде с применением Flask, SQLAlchemy, Flask-Login, форм и шаблонов, создание минимально жизнеспособного продукта (MVP) для проведения опросов или викторин с возможностью авторизации, управления вопросами, голосования и просмотра результатов, тестирование разработанного решения и подготовку к защите проекта.	Практическая работа	4
		Доработку решения кейс-задачи, расширение функциональности платформы (добавление различных типов вопросов, ограничение повторного участия, использование кодов подключения, визуализация результатов), подготовку к защите с описанием выбранной проблемы, цели, ожидаемого результата, распределения задач и возникших трудностей, оформление проекта на GitHub с README и обновлением зависимостей.	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

		Объем в ак.ч.	Объем в %
ИТОГО ПО ПРОГРАММЕ:	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

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

№ 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, СПО)	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 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. SQLiteStudio версии 3.4.0 и выше. Qt Designer. Для выполнения практических заданий требуется установленный интерпретатор Python версии 3.10 и выше, а также рекомендованная среда разработки.	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.	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://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 по Python https://docs.python.org/3/library/ — официальная дркументация по библиотекам Python https://proproprogs.ru/python — ресурс с информацией о программировании на Python https://doc.qt.io/ — официальная документация по Qt https://www.riverbankcomputing.com/static/Docs/PyQt6/ — официальная документация по PyQt6	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	При реализации программы могут использоваться следующие электронные информационные и образовательные ресурсы: https://docs.python.org/3/ — официальная документация по Python https://flask.palletsprojects.com/en/2.3.x/ — официальная документация по Flask https://docs.sqlalchemy.org/en/20/ — официальная документация по SQLAlchemy

Наименование поля	Значение полей	Значение полей	Значение полей	Значение полей
Порядковый номер модуля	1	2	3	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

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 https://leetcode.com/ — задачи по программированию	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 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://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	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

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 Маршова Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
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
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
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

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

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)
Варианты заданий	Задача 1. Сумма двух вещественных чисел с округлением до двух знаков Напиши программу, которая запрашивает два вещественных числа (каждое на отдельной строке), вычисляет их сумму и выводит результат, округлённый до двух знаков после запятой. Входные данные Два вещественных числа, каждое на отдельной строке. Выходные данные Одно вещественное число с двумя знаками после запятой. Пример Ввод: 3.1415 2.7182 Вывод: 5.86 Примечание Подсказки внутри input() не используй. Решение a = float(input()) b = float(input()) print(round(a + b, 2)) Задача 2. Площадь треугольника по формуле Герона Напиши программу, которая запрашивает длины трёх сторон треугольника (вещественные числа, каждое на отдельной строке), вычисляет его площадь по формуле Герона: S = sqrt(p * (p - a) * (p - b) * (p - c)), где p = (a + b + c) / 2. Результат округли до одного знака после запятой. Используй модуль math. Входные данные Три вещественных числа (положительные, удовлетворяющие неравенству треугольника). Выходные данные Одно вещественное число (площадь) с одним знаком после запятой. Пример Ввод: 3 4 5 Вывод: 6.0 Примечание Подсказки внутри input() не используй. Решение import math

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

```
a = float(input())
b = float(input())
c = float(input())
p = (a + b + c) / 2
area = math.sqrt(p * (p - a) * (p - b) * (p - c))
print(round(area, 1))
```

Задача 3. Сумма цифр четырёхзначного числа

Напиши программу, которая запрашивает целое четырёхзначное число (положительное) и выводит сумму его цифр.

Входные данные

Одно целое число (1000–9999).

Выходные данные

Одно целое число – сумма цифр.

Пример

Ввод:

3825

Вывод:

18

Примечание

Подсказки внутри input() не используй.

Решение

```
n = int(input())
thousands = n // 1000
hundreds = n // 100 % 10
tens = n // 10 % 10
units = n % 10
print(thousands + hundreds + tens + units)
```

Задача 4. Преобразование секунд в формат «часы:минуты:секунды»

Дано целое число T – количество секунд, прошедших с начала суток ($0 \le T < 86400$). Напиши программу, которая выводит текущее время в формате «Часы: {h}, минуты: {m}, секунды: {s}».

Дополнительное требование: если количество часов, минут или секунд меньше 10, выводить их с ведущим нулём (например, «Часы: 01»).

Входные данные

Одно целое число T.

Выходные данные

Три строки в указанном формате.

Пример

Ввод:

3665

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

	Часы: 01 минуты: 01 секунды: 05 Примечание Подсказки внутри input() не используй. Решение t = int(input()) h = t // 3600 m = (t % 3600) // 60 s = t % 60 print(f'Часы: {h:02d}') print(f'минуты: {m:02d}') print(f'секунды: {s:02d}') Задача 5. Расстояние между двумя точками на плоскости Напиши программу, которая запрашивает координаты двух точек на плоскости: x1, y1 и x2, y2 (целые числа). Вычисли евклидово расстояние по формуле: d = sqrt((x2 - x1)² + (y2 - y1)²). Результат округли до трёх знаков после запятой. Используй модуль math. Входные данные Четыре целых числа, каждое на отдельной строке: x1, y1, x2, y2. Выходные данные Одно вещественное число с тремя знаками после запятой. Пример Ввод: 0 0 3 4 Вывод: 5.000 Примечание Подсказки внутри input() не используй. Решение import math x1 = int(input()) y1 = int(input()) x2 = int(input()) y2 = int(input())
	dist = math.hypot(x2 - x1, y2 - y1) print(round(dist, 3))
Модуль 1, Основы Python, Использование ИИ-инструментов	
Количество академических часов	
Формы контроля	
Диагностические инструменты	
Показатели и критерии оценивания	
Модуль 1, Основы Python, Объектно-ориентированное программирование (ООП)	
Количество академических часов	
Формы контроля	
Диагностические инструменты	
Показатели и критерии оценивания	
Модуль 1, Основы Python, 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 с тестами 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 Мишкина Виктория Вадимовна
серийный номер 01ba319fc94f2e004e8f2fcce19aeaad
срок действия 05.12.2025 - 05.03.2027

Варианты заданий	Задача 1. Класс «Книга» (конструктор и метод)
	Создай класс Book с атрибутами: title (название), author (автор), pages (количество страниц). Реализуй:
	конструктор __init__(title, author, pages);
	метод info(), который выводит строку в формате: «{автор} — {название} ({страницы} стр.)».
	В основной программе создай экземпляр класса Book с данными, введёнными с клавиатуры (три строки: название, автор, количество страниц). Выведи результат вызова метода info().
	Входные данные
	Три строки: название, автор, количество страниц (целое число).
	Выходные данные
	Одна строка.
	Пример
	Ввод:
	Война и мир
	Толстой
	1425
	Вывод:
	Толстой — Война и мир (1425 стр.)
	Примечание
	Подсказки внутри input() не используй.
	Решение
	class Book:
	def __init__(self, title, author, pages):
	self.title = title
	self.author = author
	self.pages = pages
	
	def info(self):
	print(f'{self.author} — {self.title} ({self.pages} стр.)')
	
	title = input()
	author = input()
	pages = int(input())
	book = Book(title, author, pages)
	book.info()
	Задача 2. Класс «Студент» (статические методы и методы класса)
	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»;
	статический метод is_valid_course(course), который возвращает True, если курс от 1 до 4, иначе False;

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

метод класса from_string(data), который принимает строку в формате «Имя: курс» (например, «Анна: 3»), создаёт и возвращает объект Student;
метод __str__, возвращающий строку «Студент: {name}, курс {course}».
В основной программе прочитай одну строку в формате «Имя: курс», создай объект с помощью from_string() и выведи его строковое представление.
Входные данные
Одна строка в формате «Имя: курс».
Выходные данные
Сообщение об исправлении курса (если нужно) и строковое представление студента.
Пример
Ввод:
Пётр: 5
Вывод:
Курс исправлен на 1
Студент: Пётр, курс 1
Примечание
Подсказки внутри input() не используй.
Решение

```
class Student:
    def __init__(self, name, course):
        self.name = name
        if not self.is_valid_course(course):
            print('Курс исправлен на 1')
            self.course = 1
        else:
            self.course = course

    @staticmethod
    def is_valid_course(course):
        return 1 <= course <= 4

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

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

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 = input()
student = Student.from_string(data)
print(student)
```

Задача 3. Класс «Прямоугольник» (инкапсуляция через @property)

Создай класс Rectangle с приватными атрибутами __width и __height. Реализуй:

конструктор __init__(width, height), который принимает положительные числа; если переданное значение меньше или равно нулю, выведи «Размер должен быть больше 0» и установи значение 1;

геттеры width и height через @property, возвращающие значения;

сеттеры width и height через @setter, которые проверяют, что новое значение положительное; если нет – выведи «Размер должен быть больше 0» и не меняй значение;

метод area(), возвращающий площадь прямоугольника;

метод __str__, возвращающий строку «Прямоугольник {ширина}×{высота}».

В основной программе прочитай два целых числа (ширину и высоту). Создай объект, попробуй изменить ширину на -5 (сеттер должен вывести ошибку) и выведи площадь и строковое представление объекта.

Входные данные

Два целых числа (ширина и высота) через пробел.

Выходные данные

Сообщение об ошибке (если была), площадь, строковое представление.

Пример

Ввод:

10 5

Вывод:

Площадь: 50

Прямоугольник 10×5

Примечание

Подсказки внутри input() не используй.

Решение

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

    @property
    def width(self):
        return self.__width

    @width.setter
    def width(self, value):
        if value <= 0:
            print('Размер должен быть больше 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

```
            self.__width = 1
        else:
            self.__width = value

    @property
    def height(self):
        return self.__height

    @height.setter
    def height(self, value):
        if value <= 0:
            print('Размер должен быть больше 0')
            self.__height = 1
        else:
            self.__height = value

    def area(self):
        return self.__width * self.__height

    def __str__(self):
        return f'Прямоугольник {self.__width}×{self.__height}'

w, h = map(int, input().split())
rect = Rectangle(w, h)
rect.width = -5
print(f'Площадь: {rect.area()}')
print(rect)
```

Задача 4. Наследование: классы «Работник» и «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}».

Создай класс Manager, наследующий от Employee. Добавь атрибут bonus (премия – целое число). Переопредели метод get_info(), чтобы он возвращал строку «Менеджер: {name}, оклад: {salary}, премия: {bonus}, итого: {salary+bonus}».

В основной программе прочитай две строки: для сотрудника (имя и оклад) и для менеджера (имя, оклад, премия). Создай соответствующие объекты и выведи результат get_info() для каждого.

Входные данные

Две строки:

<имя сотрудника> <оклад>

<имя менеджера> <оклад> <премия>

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
Две строки – результаты get_info().
Пример
Ввод:
Смирнова 70000
Романов 95000 25000
Вывод:
Сотрудник: Смирнова, оклад: 70000
Менеджер: Романов, оклад: 95000, премия: 25000, итого: 120000
Примечание
Подсказки внутри input() не используй.
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}'

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}, итого: {total}'

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())
```

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. Класс «Дробь» (сравнение и арифметика)
Создай класс Fraction для работы с обыкновенными дробями. Конструктор принимает числитель и знаменатель (целые числа). Реализуй:
метод __str__, возвращающий строку «числитель/знаменатель» (сокращённую);
метод __add__ для сложения двух дробей, возвращающий новый объект Fraction (сокращённый);
метод __eq__ для сравнения двух дробей на равенство (сокращённые числители и знаменатели равны);
метод __lt__ для сравнения дробей (меньше/больше).
Для сокращения используй math.gcd.
В основной программе прочитай две строки: каждая содержит два целых числа – числитель и знаменатель. Выведи сумму дробей (в формате числитель/знаменатель) и результат сравнения (True/False) на отдельных строках.
Входные данные
Две строки, в каждой – два целых числа через пробел.
Выходные данные
Две строки: сумма дробей, результат сравнения (True или False).
Пример
Ввод:
1 3
2 5
Вывод:
11/15
False
Примечание
Подсказки внутри input() не используй.
Решение

```
import math

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

    def _reduce(self):
        gcd = math.gcd(abs(self.num), abs(self.den))
        self.num //= gcd
        self.den //= gcd
        if self.den < 0:
            self.num = -self.num
            self.den = -self.den
    def __str__(self):
        return f'{self.num}/{self.den}'

    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 __eq__(self, other):
        return self.num == other.num and self.den == other.den

    def __lt__(self, other):
        return self.num * other.den < other.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)
print(f1 == f2)
```

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191e94f2eb84c6f27ec019aedad
срок действия 05.12.2025 - 05.03.2027

	Модуль 1, Основы Python, Командный проект
Количество академических часов	
Формы контроля	
Диагностические инструменты	

Показатели и критерии оценивания	
Модуль 2, Базы данных, Алгоритмы и 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	Задача 1. Поиск элемента в связном списке (по значению)
	Дан связный список, содержащий названия городов:
	"Москва" → "Санкт-Петербург" → "Новосибирск" → "Екатеринбург" → "Казань" → "Нижний Новгород" → None
	Напиши программу, которая запрашивает у пользователя строку – название города, и выводит номер позиции (нумерация с 1), на которой этот город находится в списке. Если город не найден, выведи -1.
	Входные данные
	Одна строка – название города.
	Выходные данные
	Одно целое число (номер позиции) или -1.
	Пример
	Ввод:
	Екатеринбург
	Вывод:
	4
	Примечание
	Подсказки внутри input() не используй.
	Решение
	class Node:
	  def __init__(self, value, next_node):
	  self.value = value
	  self.next = next_node
	
	# Создаём связный список
	n6 = Node('Нижний Новгород', None)
	n5 = Node('Казань', n6)
	n4 = Node('Екатеринбург', n5)
	n3 = Node('Новосибирск', n4)
	n2 = Node('Санкт-Петербург', n3)
	head = Node('Москва', n2)
	
	target = input()
	node = head
	pos = 1
	found = -1
	while node is not None:
	  if node.value == target:
	  found = pos
	  break

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

```
    node = node.next
    pos += 1
print(found)
```

Задача 2. Вычисление выражения в обратной польской записи (с целочисленным делением)

Напиши программу, которая принимает строку – выражение в обратной польской записи, содержащее целые числа и операторы +, -, *, /. Оператор / – целочисленное деление (как в Python // для положительных, но для отрицательных должно делить в сторону минус бесконечности – используй int(a / b)). Вычисли значение, используя стек (реализуй класс Stack с методами push и pop). Результат выведи как целое число.

Входные данные

Строка, содержащая числа и операторы, разделённые пробелами.

Выходные данные

Одно целое число – результат.

Пример

Ввод:

12 3 / 2 + 4 *

Вывод:

24

(12/3=4, 4+2=6, 6*4=24)

Примечание

Подсказки внутри input() не используй.

Решение

```
class Stack:
    def __init__(self):
        self.data = []
    def push(self, val):
        self.data.append(val)
    def pop(self):
        return self.data.pop()

tokens = input().split()
stack = Stack()
for t in tokens:
    if t in ('+', '-', '*', '/'):
        b = stack.pop()
        a = stack.pop()
        if t == '+':
            stack.push(a + b)
        elif t == '-':
            stack.push(a - b)
        elif t == '*':
```

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

```
            stack.push(a * b)
        else:  # t == '/'
            # целочисленное деление с правильной обработкой отрицательных
            stack.push(int(a / b))
    else:
        stack.push(int(t))
print(stack.pop())
```

Задача 3. Класс «Очередь» на основе списка (с методом front)

Создай класс Queue, реализующий очередь с помощью списка. Методы:

push(item) – добавляет элемент в конец очереди.

pop() – удаляет и возвращает первый элемент; если очередь пуста, выводит "error" и возвращает None.

front() – возвращает первый элемент без удаления; если очередь пуста, выводит "error" и возвращает None.

size() – возвращает количество элементов.

В основной программе прочитай количество команд n, затем n строк команд:

push X – добавить X (целое число)

pop – удалить и вывести элемент

front – вывести первый элемент без удаления

size – вывести размер

Выведи результаты выполнения команд pop, front и size (каждый результат на новой строке). Гарантируется, что pop и front не будут вызываться на пустой очереди (кроме проверки пустоты – тогда они должны выводить "error").

Входные данные

Первая строка – целое число n.

Далее n строк команд.

Выходные данные

Результаты выполнения команд pop, front, size.

Пример

Ввод:

```
6
push 10
push 20
front
pop
size
pop
```

Вывод:

```
10
10
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

20
Примечание
Подсказки внутри input() не используй.
Решение

```
class Queue:
    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(0)
    def front(self):
        if not self.items:
            print('error')
            return None
        return self.items[0]
    def size(self):
        return len(self.items)

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':
        print(q.pop())
    elif cmd[0] == 'front':
        print(q.front())
    elif cmd[0] == 'size':
        print(q.size())
```

Задача 4. Сортировка вставками (по возрастанию) с выводом после каждой итерации
Реализуй сортировку вставками для списка целых чисел по возрастанию. Программа должна выводить состояние списка после каждой вставки (т.е. после того, как текущий элемент вставлен на своё место). Ввод: сначала целое число n – размер списка, затем n целых чисел через пробел. Выведи n-1 строку (для каждого шага сортировки, начиная со второго элемента). Если список уже отсортирован или состоит из одного элемента, программа всё равно должна вывести 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 – целое число n.
Вторая строка – n целых чисел через пробел.
Выходные данные
n-1 строк, каждая содержит элементы списка через пробел после очередной вставки.
Пример
Ввод:
5
4 2 7 1 3
Вывод:
2 4 7 1 3
2 4 7 1 3
1 2 4 7 3
1 2 3 4 7
Примечание
Подсказки внутри input() не используй.
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)))
```

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. Индексация с нуля.
Входные данные
Первая строка – целое число n ($1 \le n \le 10^5$).
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:
9
1 3 3 3 7 9 9 9 12
9
Вывод:
7
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)
```

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, Базы данных, Графический интерфейс (Graphical User Interface, GUI)	
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. Номеронабиратель с изменённой раскладкой
	Напиши программу на PyQt6, которая создаёт окно с кнопками, расположенными как на телефонной клавиатуре, но с изменённым порядком: цифры от 1 до 9 в три ряда по три кнопки, а в четвёртом ряду – 0, *, #. Используй QGridLayout. Установи размер кнопок 70×70 и шрифт 16. Функционал кнопок не требуется – только интерфейс.
	Входные данные:
	Нет.
	Выходные данные:
	Окно с кнопками.
	Пример (внешний вид):
	1 2 3
	4 5 6
	7 8 9
	0 * #
	Решение:
	import sys
	from PyQt6.QtWidgets import QApplication, QWidget, QGridLayout, QPushButton
	from PyQt6.QtGui import QFont
	
	class Dialer(QWidget):
	  def __init__(self):
	  super().__init__()
	  self.setWindowTitle('Номеронабиратель')
	  layout = QGridLayout()
	  keys = [
	  ['1', '2', '3'],
	  ['4', '5', '6'],
	  ['7', '8', '9'],
	  ['0', '*', '#']
	  ]
	  for row in range(4):
	  for col in range(3):
	  btn = QPushButton(keys[row][col])
	  btn.setFixedSize(70, 70)
	  btn.setFont(QFont('Arial', 16))
	  layout.addWidget(btn, row, col)
	  self.setLayout(layout)
	
	if __name__ == "__main__":
	  app = QApplication(sys.argv)

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

```
    window = Dialer()
    window.show()
    sys.exit(app.exec())
```

2. Конвертер температуры (Цельсий ↔ Фаренгейт) с QDoubleSpinBox

Создай окно с двумя полями ввода QDoubleSpinBox: одно для градусов Цельсия, другое для Фаренгейта. При изменении значения в любом поле автоматически пересчитывается значение в другом поле (используй сигнал valueChanged). Формулы: F = C * 9/5 + 32, C = (F - 32) * 5/9. Округляй результат до одного знака после запятой. В полях установи диапазон от -100 до 100.

Входные данные:

Изменение значения в QDoubleSpinBox.

Выходные данные:

Пересчитанное значение в другом поле.

Пример:

В поле °C введено 25 → в поле °F появляется 77.0.

Решение:

```
import sys
from PyQt6.QtWidgets import (QApplication, QWidget, QVBoxLayout, QHBoxLayout,
                             QLabel, QDoubleSpinBox)

class TempConverter(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle('Конвертер температур')
        layout = QVBoxLayout()

        c_layout = QHBoxLayout()
        c_layout.addWidget(QLabel('°C:'))
        self.c_spin = QDoubleSpinBox()
        self.c_spin.setRange(-100, 100)
        self.c_spin.setValue(0)
        c_layout.addWidget(self.c_spin)
        layout.addLayout(c_layout)

        f_layout = QHBoxLayout()
        f_layout.addWidget(QLabel('°F:'))
        self.f_spin = QDoubleSpinBox()
        self.f_spin.setRange(-100, 100)
        self.f_spin.setValue(32.0)
        f_layout.addWidget(self.f_spin)
        layout.addLayout(f_layout)
```

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

```
        self.setLayout(layout)

        self.c_spin.valueChanged.connect(self.c_to_f)
        self.f_spin.valueChanged.connect(self.f_to_c)

    def c_to_f(self, c):
        f = c * 9/5 + 32
        self.f_spin.blockSignals(True)
        self.f_spin.setValue(round(f, 1))
        self.f_spin.blockSignals(False)

    def f_to_c(self, f):
        c = (f - 32) * 5/9
        self.c_spin.blockSignals(True)
        self.c_spin.setValue(round(c, 1))
        self.c_spin.blockSignals(False)

if __name__ == "__main__":
    app = QApplication(sys.argv)
    window = TempConverter()
    window.show()
    sys.exit(app.exec())
```

3. Генератор паролей с QSpinBox и QCheckBox

Напиши программу, которая генерирует случайный пароль заданной длины (от 4 до 16 символов). Пользователь выбирает длину с помощью QSpinBox и отмечает QCheckBox: «Включить цифры» и «Включить спецсимволы» (набор !@#$%^&*). По умолчанию включены оба флажка. При нажатии на кнопку «Сгенерировать» в поле QLineEdit (только для чтения) появляется пароль из букв латинского алфавита, а также, в зависимости от выбора, цифр и/или спецсимволов.

Входные данные:

Значения спинбокса, состояние чекбоксов, нажатие кнопки.

Выходные данные:

Строка с паролем в QLineEdit.

Пример:

Длина = 10, включены цифры, отключены символы → вывод aB4fG7hJkL.

Решение:

```
import sys
import random
import string
from PyQt6.QtWidgets import (QApplication, QWidget, QVBoxLayout, QHBoxLayout,
                             QLabel, QSpinBox, QCheckBox, QPushButton, QLineEdit)
```

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 PasswordGen(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle('Генератор паролей')
        layout = QVBoxLayout()

        len_layout = QHBoxLayout()
        len_layout.addWidget(QLabel('Длина:'))
        self.length_spin = QSpinBox()
        self.length_spin.setRange(4, 16)
        self.length_spin.setValue(8)
        len_layout.addWidget(self.length_spin)
        layout.addLayout(len_layout)

        self.digits_check = QCheckBox('Включить цифры')
        self.digits_check.setChecked(True)
        layout.addWidget(self.digits_check)

        self.symbols_check = QCheckBox('Включить спецсимволы (!@#$%^&*)')
        self.symbols_check.setChecked(True)
        layout.addWidget(self.symbols_check)

        self.btn = QPushButton('Сгенерировать')
        self.btn.clicked.connect(self.generate)
        layout.addWidget(self.btn)

        self.result_edit = QLineEdit()
        self.result_edit.setReadOnly(True)
        layout.addWidget(self.result_edit)

        self.setLayout(layout)

    def generate(self):
        length = self.length_spin.value()
        chars = string.ascii_letters
        if self.digits_check.isChecked():
```

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

```
            chars += string.digits
        if self.symbols_check.isChecked():
            chars += '!@#$%^&*'
        password = ''.join(random.choice(chars) for _ in range(length))
        self.result_edit.setText(password)

if __name__ == "__main__":
    app = QApplication(sys.argv)
    window = PasswordGen()
    window.show()
    sys.exit(app.exec())
```

4. Калькулятор целых чисел с выбором операции

Создай окно с двумя полями ввода QLineEdit для целых чисел, выпадающим списком QComboBox с операциями (+, -, *, /), кнопкой «=» и полем для вывода результата QLineEdit (только для чтения). При нажатии на кнопку выполняется соответствующая операция. Используй QIntValidator для полей ввода (только целые числа). При делении на ноль выводить «Ошибка». Используй менеджеры макетов QVBoxLayout и QHBoxLayout.

Входные данные:

Два целых числа, выбор операции.

Выходные данные:

Результат в поле вывода.

Пример:

В поле1 = 10, операция = *, поле2 = 5 → результат 50.

Решение:

```
import sys
from PyQt6.QtGui import QIntValidator
from PyQt6.QtWidgets import (QApplication, QWidget, QVBoxLayout, QHBoxLayout,
                             QLineEdit, QComboBox, QPushButton)

class Calculator(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle('Калькулятор')
        layout = QVBoxLayout()

        row1 = QHBoxLayout()
        self.num1 = QLineEdit()
        self.num1.setValidator(QIntValidator())
        row1.addWidget(self.num1)
        self.operation = QComboBox()
        self.operation.addItems(['+', '-', '*', '/'])
```

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

```
        row1.addWidget(self.operation)
        self.num2 = QLineEdit()
        self.num2.setValidator(QIntValidator())
        row1.addWidget(self.num2)
        layout.addLayout(row1)

        self.btn = QPushButton('=')
        self.btn.clicked.connect(self.calculate)
        layout.addWidget(self.btn)

        self.result = QLineEdit()
        self.result.setReadOnly(True)
        layout.addWidget(self.result)

        self.setLayout(layout)

    def calculate(self):
        try:
            a = int(self.num1.text() or '0')
            b = int(self.num2.text() or '0')
            op = self.operation.currentText()
            if op == '+':
                res = a + b
            elif op == '-':
                res = a - b
            elif op == '*':
                res = a * b
            elif op == '/':
                if b == 0:
                    raise ZeroDivisionError
                res = a / b
            self.result.setText(str(res))
        except ZeroDivisionError:
            self.result.setText('Ошибка')
        except ValueError:
            self.result.setText('Ошибка')
```

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 __name__ == "__main__":
	app = QApplication(sys.argv)
	window = Calculator()
	window.show()
	sys.exit(app.exec())
	5. Форма авторизации с QFormLayout
	Создай окно с полями «Логин» и «Пароль» (используй QLineEdit, пароль скрыт через setEchoMode(QLineEdit.EchoMode.Password)), а также кнопку «Войти».
	При нажатии на кнопку появляется QMessageBox с текстом «Вход выполнен» (реальная проверка не требуется). Используй QFormLayout для размещения полей.
	Входные данные:
	Ввод текста в поля (не влияет на результат).
	Выходные данные:
	Окно с QMessageBox.
	Пример:
	После нажатия кнопки появляется сообщение «Вход выполнен».
	Решение:
	import sys
	from PyQt6.QtWidgets import (QApplication, QWidget, QFormLayout,
	QLineEdit, QPushButton, QMessageBox)
	
	class LoginForm(QWidget):
	def __init__(self):
	super().__init__()
	self.setWindowTitle('Авторизация')
	layout = QFormLayout()
	
	self.login_edit = QLineEdit()
	self.password_edit = QLineEdit()
	self.password_edit.setEchoMode(QLineEdit.EchoMode.Password)
	
	self.btn = QPushButton('Войти')
	self.btn.clicked.connect(self.login)
	
	layout.addRow('Логин:', self.login_edit)
	layout.addRow('Пароль:', self.password_edit)
	layout.addRow(self.btn)
	
	self.setLayout(layout)

	def login(self):
	QMessageBox.information(self, 'Успех', 'Вход выполнен')
	
	if __name__ == "__main__":
	app = QApplication(sys.argv)
	window = LoginForm()
	window.show()
	sys.exit(app.exec())

Модуль 2, Базы данных, Базы 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)

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. Выборка фильмов с условием и сортировкой Напиши SQL-запрос, который выводит названия и длительность всех фильмов из таблицы movies, у которых год выпуска больше 2010. Результат отсортируй по убыванию длительности (самые длинные сверху). Таблицы movies имеет поля: id (INTEGER PRIMARY KEY) title (TEXT) year (INTEGER) duration (INTEGER) director_id (INTEGER – внешний ключ к directors) Решение SELECT title, duration FROM movies WHERE year > 2010 ORDER BY duration DESC; Задача 2. Добавление нового режиссёра Напиши SQL-запрос, который добавляет в таблицу directors нового режиссёра с именем «Алексей Балабанов». Таблицы directors содержит поля: id (INTEGER PRIMARY KEY AUTOINCREMENT) name (TEXT) Решение INSERT INTO directors(name) VALUES ('Алексей Балабанов'); Задача 3. Обновление длительности фильмов Напиши SQL-запрос, который увеличивает длительность всех фильмов режиссёра с id = 3 на 10 минут. Таблицы movies содержит поле duration. Решение UPDATE movies SET duration = duration + 10 WHERE director_id = 3; Задача 4. Удаление фильмов без жанров Напиши SQL-запрос, который удаляет все фильмы, у которых нет ни одного жанра (т.е. отсутствуют записи в таблице связей movies_genres). Используй подзапрос или NOT EXISTS. Таблицы movies (id, ...) movies_genres (movie_id, genre_id) Решение

	DELETE FROM movies
	WHERE NOT EXISTS (
	  SELECT 1
	  FROM movies_genres
	  WHERE movies_genres.movie_id = movies.id
	);
	Задача 5. Количество фильмов по каждому режиссёру (с JOIN и GROUP BY)
	Напиши SQL-запрос, который выводит имя режиссёра и количество фильмов, снятых этим режиссёром. Результат отсортируй по убыванию количества.
	Используй таблицы directors и movies, соединив их по director_id. Название столбца с количеством должно быть film_count.
	Таблицы
	directors (id, name)
	movies (id, title, year, duration, director_id)
	Решение
	SELECT directors.name, COUNT(movies.id) AS film_count
	FROM directors
	LEFT JOIN movies ON directors.id = movies.director_id
	GROUP BY directors.id
	ORDER BY film_count DESC;

Модуль 2, Базы данных, Командный проект	
Количество академических часов	
Формы контроля	
Диагностические инструменты	
Показатели и критерии оценивания	

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Маршеня Виктория Вадимовна
серийный номер 01ba3191e94f2eb04e6f27ee019acdad
срок действия 05.12.2025 - 05.03.2027

Модуль 3, 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)

Варианты заданий	Задача 1. Сумма положительных чисел из файла
	В файле numbers.txt хранятся целые числа, разделённые пробелами (в нескольких строках). Напиши программу, которая вычисляет сумму только положительных чисел (больше нуля) из этого файла. Ноль и отрицательные числа игнорируются. Выведи результат.
	Входные данные (файл numbers.txt)
	Пример содержимого:
	-5 10 3 -2
	0 7 8
	-1 4
	Выходные данные
	Одно целое число – сумма положительных чисел.
	Пример вывода
	32
	(10+3+7+8+4 = 32)
	Примечание
	Файл должен быть в той же директории, что и программа. Кодировка – UTF-8.
	Решение
	total = 0
	with open('numbers.txt', encoding='utf-8') as f:
	  for line in f:
	  for num in line.split():
	  n = int(num)
	  if n > 0:
	  total += n
	print(total)
	Задача 2. Максимальная температура из JSON-файла
	Скачай файл weather.json (содержит данные о погоде за несколько дней с шагом 3 часа). Напиши программу, которая находит максимальную температуру (temp_max) за весь период и выводит её. Используй модуль json.
	Входные данные (файл weather.json)
	Структура файла:
	{
	  "list": [
	  {"main": {"temp_min": 5.2, "temp_max": 7.8}},
	  {"main": {"temp_min": -1.3, "temp_max": 2.5}},
	  ...
	  ]
	}
	Выходные данные
	Одно вещественное число (максимальная температура).
	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

7.8
Решение

```
import json
with open('weather.json', encoding='utf-8') as f:
    data = json.load(f)
max_temp = float('-inf')
for entry in data['list']:
    t = entry['main']['temp_max']
    if t > max_temp:
        max_temp = t
print(max_temp)
```

Задача 3. Персонаж с минимальным ростом из CSV-файла
В файле characters.csv в формате CSV есть столбцы name и height (рост в сантиметрах). Некоторые строки могут содержать 'NA' (неизвестно). Напиши программу, которая находит имя персонажа с наименьшим ростом (среди тех, у кого рост указан). Если несколько персонажей имеют одинаковый минимальный рост, выведи того, кто встречается первым в файле.
Входные данные (файл characters.csv)
Пример содержимого:

```
name,height
Luke Skywalker,172
Darth Vader,202
Yoda,66
R2-D2,NA
```

Выходные данные
Имя персонажа (строка).
Пример вывода

```
Yoda
```

Решение

```
import csv
min_height = float('inf')
min_name = ''
with open('characters.csv', newline='', encoding='utf-8') as f:
    reader = csv.DictReader(f)
    for row in reader:
        if row['height'] != 'NA':
            h = int(row['height'])
            if h < min_height:
                min_height = h
                min_name = row['name']
print(min_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

Задача 4. Группировка файлов по расширению (sys.stdin)
С клавиатуры (через sys.stdin) вводится произвольное количество строк, каждая строка содержит имя файла в формате «имя.расширение». Напиши программу, которая группирует файлы по расширению, затем для каждой группы выводит (в алфавитном порядке расширений):
список имён файлов (только имена, без расширения) в алфавитном порядке,
строку «Количество файлов {расширение}: {количество}»,
пустую строку после каждой группы.
Входные данные
Строки вида имя.расширение (например, readme.txt, script.py, data.csv). Конец ввода – Ctrl+D (или конец файла).
Выходные данные
См. формат в примере.
Пример ввода
document.txt
image.jpg
note.txt
photo.jpg
main.py
Пример вывода
image
photo
Количество файлов jpg: 2

main
Количество файлов py: 1

document
note
Количество файлов txt: 2
Решение

```
import sys
groups = {}
for line in sys.stdin:
    name, ext = line.strip().split('.')
    groups.setdefault(ext, []).append(name)
for ext in sorted(groups):
    for name in sorted(groups[ext]):
        print(name)
    print(f'Количество файлов {ext}: {len(groups[ext])}')
    print()
```

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. Исправление ошибок в коде (поиск числа в списке)
	Дан код, который должен проверять, встречается ли заданное число в списке numbers. Однако в коде есть ошибки (логическая и возможно синтаксическая). Найди и исправь их, чтобы программа работала корректно для любого ввода.
	Исходный код с ошибками
	numbers = [3, 5, 7, 9, 11, 13]
	found = False
	x = int(input())
	for i in range(len(numbers) - 1):
	  if numbers[i] == x:
	  found = True
	if found:
	  print('Число найдено')
	else:
	  print('Число не найдено')
	Входные данные
	Одно целое число x.
	Выходные данные
	Строка «Число найдено» или «Число не найдено».
	Примеры
	Ввод:
	7
	Вывод:
	Число найдено
	Ввод:
	8
	Вывод:
	Число не найдено
	Решение (исправленный код)
	numbers = [3, 5, 7, 9, 11, 13]
	found = False
	x = int(input())
	for i in range(len(numbers)): # исправлено: len(numbers) вместо len(numbers)-1
	  if numbers[i] == x:
	  found = True
	  break # добавлено: досрочный выход при нахождении
	if found:
	  print('Число найдено')
	else:
	  print('Число не найдено')

Модуль 3, WEB и API, Ответственное использование ИИ-инструментов	
Количество академических часов	
Формы контроля	
Диагностические инструменты	
Показатели и критерии оценивания	
Модуль 3, 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 Марцинив Виктория Вадимовна

серийный номер 01ba3191c94f2cb04c6f27cc019aedad

срок действия 05.12.2025 - 05.03.2027

Варианты заданий	1. Калькулятор с операциями и флагом округления (argparse)
	Напиши программу, которая принимает два целых числа как позиционные аргументы и необязательный ключ -op с выбором операции: add, sub, mul, div (по умолчанию add). Добавь ключ --round (флаг), который округляет результат до целого (если указан). При делении на ноль или неверной операции выводить «Error». Используй модуль argparse.
	Примеры работы:
	$ python calc.py 10 3
	13
	$ python calc.py 10 3 -op mul
	30
	$ python calc.py 10 3 -op div --round
	3
	$ python calc.py 10 0 -op div
	Error
	Входные данные:
	Аргументы командной строки.
	Выходные данные:
	Число или «Error».
	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('-op', choices=['add','sub','mul','div'], default='add')
	parser.add_argument('--round', action='store_true')
	args = parser.parse_args()
	
	try:
	res = ops[args.op](args.a, args.b)
	if args.round:
	print(round(res))
	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(res)
except ZeroDivisionError:
    print('Error')
```

2. Количество дней до следующего дня рождения (datetime)

Напиши программу, которая запрашивает у пользователя дату рождения в формате ДД.ММ.ГГГГ и выводит количество полных лет и количество дней до следующего дня рождения (в днях). Если день рождения уже прошёл в текущем году, считается следующий год. Используй модуль datetime. Вывод оформи так: «Вам N лет. До следующего дня рождения осталось D дней.»

Входные данные:

Одна строка – дата рождения.

Выходные данные:

Одна строка с количеством лет и дней.

Пример ввода (сегодня 01.01.2026):

15.03.2000

Пример вывода:

Вам 25 лет. До следующего дня рождения осталось 73 дней.

Решение:

```
from datetime import datetime, date

birth = datetime.strptime(input(), '%d.%m.%Y').date()
today = date.today()
age = today.year - birth.year
if (today.month, today.day) < (birth.month, birth.day):
    age -= 1
# следующий день рождения
next_birth = date(today.year, birth.month, birth.day)
if next_birth < today:
    next_birth = date(today.year + 1, birth.month, birth.day)
delta = (next_birth - today).days
print(f'Вам {age} лет. До следующего дня рождения осталось {delta} дней.')
```

3. Поиск всех email-адресов в тексте (регулярные выражения)

Напиши программу, которая читает строку текста и выводит все найденные email-адреса (каждый на новой строке) в порядке их появления. Email считается корректным, если он содержит @, до и после которого есть как минимум один символ из букв, цифр, точки или дефиса. Используй модуль re.

Входные данные:

Одна строка текста.

Выходные данные:

Все email-адреса, каждый на новой строке.

Пример ввода:

Контакты: admin@site.com, support@mail.ru, и test@yandex.ru

Пример вывода:

admin@site.com

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

support@mail.ru
test@yandex.ru
Решение:

```
import re
text = input()
emails = re.findall(r'[\w.-]+@[\w.-]+', text)
for email in emails:
    print(email)
```

4. Функция с doctest и unittest для нахождения уникальных чисел
Напиши функцию unique_numbers(lst), которая принимает список чисел и возвращает список чисел, встречающихся ровно один раз, в порядке их первого появления. Добавь докстринг с примерами (doctest). Также напиши unittest-тесты для этой функции (в отдельном файле или том же). В основной программе считай список чисел с клавиатуры и выведи результат через пробел.
Входные данные:
Строка чисел через пробел.
Выходные данные:
Числа, встречающиеся один раз, через пробел.
Пример ввода:
1 2 2 3 4 4 5
Пример вывода:
1 3 5
Решение:

```
def unique_numbers(lst):
    """
    >>> unique_numbers([1, 2, 2, 3, 4, 4, 5])
    [1, 3, 5]
    >>> unique_numbers([1, 1, 2])
    [2]
    >>> unique_numbers([1, 2, 3])
    [1, 2, 3]
    """
    from collections import Counter
    counts = Counter(lst)
    return [x for x in lst if counts[x] == 1]

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

# unittest (можно вынести в отдельный файл)
```

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 unittest
	from main import unique_numbers
	class TestUnique(unittest.TestCase):
	  def test_mixed(self):
	  self.assertEqual(unique_numbers([1,2,2,3,4,4,5]), [1,3,5])
	  def test_empty(self):
	  self.assertEqual(unique_numbers([]), [])
	  def test_all_unique(self):
	  self.assertEqual(unique_numbers([5,4,3]), [5,4,3])
	if __name__ == '__main__':
	  unittest.main()
	"""
	В основной программе:
	lst = list(map(int, input().split()))
	print(*unique_numbers(lst))
	5. Система логирования попыток входа (logging)
	Напиши программу, которая имитирует систему входа с проверкой доступа. В программе задан словарь users = {"admin": "1234", "user": "qwerty"}. Программа запрашивает логин и пароль (по отдельным строкам). Если логин не найден, выводится «Пользователь не найден» и в лог записывается ошибка уровня ERROR. Если пароль неверен, выводится «Неверный пароль» и в лог – предупреждение WARNING. Если вход успешен, выводится «Добро пожаловать» и в лог – INFO. Логи должны записываться в файл login.log с форматом: %(asctime)s - %(levelname)s - %(message)s. Обработай возможные исключения с try/except и запиши их в лог с logging.exception().
	Входные данные:
	Две строки: логин и пароль.
	Выходные данные:
	Сообщение о результате.
	Пример ввода:
	admin
	1234
	Пример вывода:
	Добро пожаловать
	Решение:
	import logging
	
	logging.basicConfig(
	  level=logging.INFO,
	  filename='login.log',
	  encoding='utf-8',
	  format='%(asctime)s - %(levelname)s - %(message)s'
	)

	users = {"admin": "1234", "user": "qwerty"}
	
	try:
	  login = input()
	  password = input()
	  if login not in users:
	  print('Пользователь не найден')
	  logging.error(f'Попытка входа неизвестного пользователя: {login}')
	  elif users[login] != password:
	  print('Неверный пароль')
	  logging.warning(f'Неверный пароль для пользователя: {login}')
	  else:
	  print('Добро пожаловать')
	  logging.info(f'Успешный вход пользователя: {login}')
	except Exception:
	  logging.exception('Ошибка при выполнении программы')

Модуль 3, 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

Контур Крипто
владелец Маршева 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. 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. Количество вакансий разработчиков в Москве (API) Напиши программу, которая делает запрос к API «Работа России» по адресу: http://opendata.trudvsem.ru/api/v1/vacancies/region/77 (регион 77 – Москва) с параметром text равным «разработчик». Из полученного JSON-ответа извлеки количество вакансий (поле results.vacancies – это список, количество его элементов и есть ответ). Выведи полученное число. Входные данные Нет (регион и текст запроса фиксированы). Выходные данные Одно целое число – количество найденных вакансий. Пример вывода 42 Решение import requests url = 'http://opendata.trudvsem.ru/api/v1/vacancies/region/77' params = {'text': 'разработчик'} response = requests.get(url, params=params) data = response.json() vacancies = data['results']['vacancies'] print(len(vacancies)) Задача 2. Подсчёт количества Python Enhancement Proposals (PEP) в секции «Accepted» Напиши программу, которая парсит страницу https://peps.python.org/ и подсчитывает количество PEP в секции с идентификатором accepted-peps (тег <section id="accepted-peps">). Внутри этой секции все PEP перечислены в таблице <table>. Подсчитай количество строк <tr> в этой таблице и выведи полученное число. Входные данные Нет. Выходные данные Одно целое число. Пример вывода 9 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='accepted-peps')

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

```
table = section.find('table')
rows = table.find_all('tr')
print(len(rows))
```

Задача 3. Фильм с наибольшим рейтингом (CSV)

В файле ratings.csv содержатся данные о фильмах в формате CSV с колонками: Title, Genres, Rating. Напиши программу, которая находит фильм с самым высоким рейтингом (поле Rating – вещественное число). Если таких фильмов несколько, выведи первый по порядку в файле. Выведи строку в формате: «Название: {название}, рейтинг: {рейтинг}».

Входные данные (файл ratings.csv)

Пример содержимого:

```
Title,Genres,Rating
The Shawshank Redemption,Drama,9.3
The Godfather,Crime|Drama,9.2
The Dark Knight,Action|Crime,9.0
```

Выходные данные

Одна строка с названием и рейтингом.

Пример вывода

```
Название: The Shawshank Redemption, рейтинг: 9.3
```

Решение

```
import csv

best_title = ''
best_rating = -1.0
with open('ratings.csv', newline='', encoding='utf-8') as f:
    reader = csv.DictReader(f)
    for row in reader:
        rating = float(row['Rating'])
        if rating > best_rating:
            best_rating = rating
            best_title = row['Title']
print(f'Название: {best_title}, рейтинг: {best_rating}')
```

Задача 4. Курс доллара к рублю (JSON)

В файле rates.json содержится JSON-структура с курсами валют, например, как в ответе ЦБ РФ. У валюты USD (доллар США) есть поле Value – курс к рублю. Напиши программу, которая читает файл rates.json и выводит текущий курс доллара (число с плавающей точкой) в формате: «Курс USD: {курс} руб.».

Входные данные (файл rates.json)

Пример структуры:

```
{
  "Valute": {
    "USD": {
```

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

```
    "CharCode": "USD",
    "Name": "Доллар США",
    "Value": 72.3456
   },
   "EUR": {
    "CharCode": "EUR",
    "Value": 85.1234
   }
  }
}
```


Выходные данные
Одна строка с курсом.
Пример вывода
Курс USD: 72.3456 руб.
Решение

```
import json

with open('rates.json', encoding='utf-8') as f:
    data = json.load(f)
usd_value = data['Valute']['USD']['Value']
print(f'Курс USD: {usd_value} руб.')
```

Задача 5. Калькулятор с операциями и аргументами командной строки
Напиши программу calc_advanced.py, которая принимает два целых числа (позиционные аргументы) и опциональный ключ --op (или -o) с возможными значениями: add, sub, mul, pow.
Если add – вывести сумму.
sub – разность (первое минус второе).
mul – произведение.
pow – первое число, возведённое в степень второго.
Если ключ не указан, по умолчанию – сложение.
В случае ошибки (например, деление на ноль – но у нас нет деления, или отрицательная степень – возведение в степень работает с отрицательными) вывести «Ошибка».
Примеры вызова и вывода

```
$ python calc_advanced.py 5 3
8

$ python calc_advanced.py 5 3 -o mul
15

$ python calc_advanced.py 2 5 -o pow
32
```

Решение

```
import argparse

ops = {
    'add': lambda a, b: a + b,
    'sub': lambda a, b: a - b,
    'mul': lambda a, b: a * b,
    'pow': 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()

try:
    result = ops[args.op](args.a, args.b)
    print(result)
except Exception:
    print('Ошибка')
```


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

Количество академических часов	

Формы контроля	
Диагностические инструменты	
Показатели и критерии оценивания	
Модуль 4, Фреймворк Flask, Модуль Flask	
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. Произведение двух чисел через динамический URL
	Напиши Flask-приложение, которое по адресу /multiply/<int:a>/<int:b> выводит HTML-страницу с результатом умножения a * b. Используй динамический маршрут с двумя целочисленными параметрами. Функция должна возвращать строку с HTML-разметкой, содержащую результат.
	Входные данные:
	Два целых числа в URL (например, /multiply/4/5).
	Выходные данные:
	Страница с текстом (например, «Результат умножения: 20»).
	Пример:
	При переходе по адресу /multiply/3/7 на странице отображается:
	Результат умножения: 21
	Решение:
	from flask import Flask
	
	app = Flask(__name__)
	
	@app.route('/multiply/<int:a>/<int:b>')
	def multiply(a, b):
	  result = a * b
	  return f' Результат умножения: {result} '
	
	if __name__ == '__main__':
	  app.run(debug=True)
	2. Страница со списком книг с использованием шаблона
	Напиши Flask-приложение, которое по адресу /books отображает список книг, переданный через render_template(). Создай шаблон books.html в папке templates, который выводит заголовок «Список книг» и список ( ) с названиями книг, используя цикл Jinja2. Переменная books – список строк.
	Входные данные:
	Нет (данные заданы в коде Python).
	Выходные данные:
	HTML-страница с заголовком и списком книг.
	Пример:
	При переходе на /books отображается:
	Список книг
	- Война и мир
	- Преступление и наказание
	- Мастер и Маргарита
	Решение:
	main.py:
	from flask import Flask, render_template

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

```
app = Flask(__name__)

books = ['Война и мир', 'Преступление и наказание', 'Мастер и Маргарита']

@app.route('/books')
def show_books():
    return render_template('books.html', books=books)

if __name__ == '__main__':
    app.run(debug=True)
templates/books.html:
html
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Список книг</title>
</head>
<body>
    <h1>Список книг</h1>
    <ul>
    {% for book in books %}
        <li>{{ book }}</li>
    {% endfor %}
    </ul>
</body>
</html>
```

3. Страница с облаком категорий (использование фильтра и подсчёт)

Напиши Flask-приложение, которое по адресу /categories отображает список категорий из словаря {'Новости': 5, 'Спорт': 3, 'Музыка': 4} в виде маркированного списка. Для вывода используй шаблон, а в нём – цикл по items(). Для каждой категории выведи название и количество материалов в формате «Категория (количество)». Используй фильтр |length или просто выведи значение.

Входные данные:

Нет (данные заданы в коде Python).

Выходные данные:

HTML-страница со списком категорий.

Пример:

На странице отображается:

Категории

- Новости (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

- Спорт (3)
- Музыка (4)
Решение:
main.py:
```
from flask import Flask, render_template

app = Flask(__name__)

categories = {'Новости': 5, 'Спорт': 3, 'Музыка': 4}

@app.route('/categories')
def show_categories():
    return render_template('categories.html', categories=categories)

if __name__ == '__main__':
    app.run(debug=True)
```
templates/categories.html:
html
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Категории</title>
</head>
<body>
    <h1>Категории</h1>
    <ul>
    {% for name, count in categories.items() %}
        <li>{{ name }} ({{ count }})</li>
    {% endfor %}
    </ul>
</body>
</html>
```
4. Форма для добавления рецепта (с валидацией и перенаправлением)
Напиши Flask-приложение с формой для добавления рецепта. Используй библиотеку Flask-WTF. Форма должна содержать поля: название (обязательное, длина до 100 символов) и описание (обязательное). После успешной отправки формы рецепт добавляется в список recipes (глобальный список), и пользователь перенаправляется на страницу со списком рецептов (маршрут /recipes). Страница со списком должна выводить все рецепты в виде карточек (заголовок и краткое описание, обрезанное до 50 символов через фильтр |truncate(50)).
Входные данные:
Заполненная форма (название и описание).

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:
Перенаправление на страницу со списком рецептов.
Пример:
После добавления рецепта «Борщ» с описанием «Сварить свёклу...» на странице /recipes появляется карточка с этим рецептом.
Решение:
main.py:

```
from flask import Flask, render_template, redirect, url_for
from flask_wtf import FlaskForm
from wtforms import StringField, TextAreaField, SubmitField
from wtforms.validators import DataRequired, Length

app = Flask(__name__)
app.config['SECRET_KEY'] = 'mysecretkey'

recipes = []

class RecipeForm(FlaskForm):
    title = StringField('Название', validators=[DataRequired(), Length(max=100)])
    description = TextAreaField('Описание', validators=[DataRequired()])
    submit = SubmitField('Добавить')

@app.route('/recipes')
def show_recipes():
    return render_template('recipes.html', recipes=recipes)

@app.route('/add_recipe', methods=['GET', 'POST'])
def add_recipe():
    form = RecipeForm()
    if form.validate_on_submit():
        recipes.append({'title': form.title.data, 'description': form.description.data})
        return redirect(url_for('show_recipes'))
    return render_template('add_recipe.html', form=form)

if __name__ == '__main__':
    app.run(debug=True)
```

templates/base.html (базовый шаблон для наследования):
html

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

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>{% block title %}Рецепты{% endblock %}</title>
</head>
<body>
    <div class="container">
        {% block content %}{% endblock %}
    </div>
</body>
</html>
templates/recipes.html:
html
{% extends "base.html" %}

{% block title %}Список рецептов{% endblock %}

{% block content %}
<h1>Список рецептов</h1>
<a href="{{ url_for('add_recipe') }}">Добавить рецепт</a>
{% for recipe in recipes %}
    <div class="card">
        <h3>{{ recipe.title }}</h3>
        <p>{{ recipe.description|truncate(50) }}</p>
    </div>
{% else %}
    <p>Рецептов пока нет.</p>
{% endfor %}
{% endblock %}
templates/add_recipe.html:
html
{% extends "base.html" %}

{% block title %}Добавить рецепт{% endblock %}

{% block content %}
```

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

	Добавить рецепт
	<form method="POST">
	{{ form.csrf_token }}
	{{ form.title.label }} {{ form.title() }}
	{{ form.description.label }} {{ form.description() }}
	{{ form.submit() }}
	</form>
	Назад к списку
	{% endblock %}
	5. Маршрут с выводом текущей даты и времени
	Напиши Flask-приложение с единым маршрутом /info/<type>, где type может быть 'date' или 'time'. Если type='date' – отобразить текущую дату в формате ДД.ММ.ГГГГ. Если type='time' – отобразить текущее время в формате ЧЧ:ММ. Если передан другой параметр – вывести сообщение «Неизвестный тип». Используй один обработчик для обоих случаев. Для получения даты и времени используй модуль datetime.
	Входные данные:
	Строка в URL (/info/date или /info/time или /info/anything).
	Выходные данные:
	Страница с датой, временем или сообщением об ошибке.
	Пример:
	При переходе на /info/date отображается 15.06.2026.
	При переходе на /info/time отображается 14:30.
	При переходе на /info/unknown отображается Неизвестный тип.
	Решение:
	import datetime
	from flask import Flask
	
	app = Flask(__name__)
	
	@app.route('/info/<string:type>')
	def info(type):
	now = datetime.datetime.now()
	if type == 'date':
	return f' {now.strftime("%d.%m.%Y")} '
	elif type == 'time':
	return f' {now.strftime("%H:%M")} '
	else:
	return ' Неизвестный тип '
	
	if __name__ == '__main__':
	app.run(debug=True)

Модуль 4, Фреймворк Flask, SQLAlchemy	
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. Модель и форма для добавления книг
	Создай Flask-приложение с базой данных SQLite (через SQLAlchemy) и моделью Book с полями: id (первичный ключ), title (строка, обязательное), author (строка, обязательное), year (целое число, необязательное), rating (целое число, по умолчанию 0). Создай форму BookForm с полями title, author, year, rating и кнопкой отправки. На главной странице (/) выведи список всех книг в виде таблицы (колонки: ID, Название, Автор, Год, Рейтинг) и форму для добавления новой книги. После успешной отправки формы книга добавляется в БД и страница обновляется (перенаправление на /). Для валидации: title и author обязательны, year – опционально (можно использовать Optional()).
	Входные данные:
	Данные формы (название, автор, год, рейтинг).
	Выходные данные:
	Страница со списком всех книг и формой.
	Пример:
	После добавления книги «Война и мир» автора «Толстой» с годом 1869, она появляется в списке.
	Решение:
	from flask import Flask, render_template, redirect, url_for
	from flask_sqlalchemy import SQLAlchemy
	from flask_wtf import FlaskForm
	from wtforms import StringField, IntegerField, SubmitField
	from wtforms.validators import DataRequired, Optional
	
	app = Flask(__name__)
	app.config['SECRET_KEY'] = 'secret'
	app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///books.db'
	db = SQLAlchemy(app)
	
	class Book(db.Model):
	id = db.Column(db.Integer, primary_key=True)
	title = db.Column(db.String(255), nullable=False)
	author = db.Column(db.String(255), nullable=False)
	year = db.Column(db.Integer)
	rating = db.Column(db.Integer, default=0)
	
	db.create_all()
	
	class BookForm(FlaskForm):
	title = StringField('Название', validators=[DataRequired()])
	author = StringField('Автор', validators=[DataRequired()])
	year = IntegerField('Год', validators=[Optional()])
	rating = IntegerField('Рейтинг', default=0)
	submit = SubmitField('Добавить')
	
	@app.route('/', methods=['GET', 'POST'])

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 index():
    form = BookForm()
    if form.validate_on_submit():
        book = Book(
            title=form.title.data,
            author=form.author.data,
            year=form.year.data,
            rating=form.rating.data
        )
        db.session.add(book)
        db.session.commit()
        return redirect(url_for('index'))
    books = Book.query.all()
    return render_template('index.html', form=form, books=books)

if __name__ == '__main__':
    app.run(debug=True)
```

(Шаблон index.html должен выводить форму и таблицу книг – в рамках текущего контроля достаточно привести только Python-код.)

2. Модели категорий и товаров с отображением по категориям

Создай Flask-приложение с двумя моделями: Category (поля: id, title) и Product (поля: id, name, price, category_id – внешний ключ к Category.id). Реализуй связь «один ко многим». Создай форму ProductForm для добавления товара с полями name (обязательное), price (обязательное), category – выпадающий список (SelectField) с выбором категории. На главной странице выведи список всех товаров с указанием категории. Также добавь маршрут /category/<int:id>, который показывает товары только выбранной категории и её название. Форма должна быть доступна на главной.

Входные данные:

Данные формы (название товара, цена, категория).

Выходные данные:

Список товаров (всех или по категории).

Пример:

При добавлении товара «Ноутбук» в категорию «Электроника», он появляется в списке, а при переходе по /category/1 видны только товары этой категории.

Решение:

```
from flask import Flask, render_template, redirect, url_for
from flask_sqlalchemy import SQLAlchemy
from flask_wtf import FlaskForm
from wtforms import StringField, IntegerField, SelectField, SubmitField
from wtforms.validators import DataRequired

app = Flask(__name__)
app.config['SECRET_KEY'] = 'secret'
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///shop.db'
db = SQLAlchemy(app)
```

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 Category(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    title = db.Column(db.String(255), nullable=False)
    products = db.relationship('Product', back_populates='category')

class Product(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(255), nullable=False)
    price = db.Column(db.Integer, nullable=False)
    category_id = db.Column(db.Integer, db.ForeignKey('category.id'))
    category = db.relationship('Category', back_populates='products')

db.create_all()

class ProductForm(FlaskForm):
    name = StringField('Название', validators=[DataRequired()])
    price = IntegerField('Цена', validators=[DataRequired()])
    category = SelectField('Категория', coerce=int)
    submit = SubmitField('Добавить')

@app.route('/', methods=['GET', 'POST'])
def index():
    form = ProductForm()
    form.category.choices = [(c.id, c.title) for c in Category.query.all()]
    if form.validate_on_submit():
        product = Product(
            name=form.name.data,
            price=form.price.data,
            category_id=form.category.data
        )
        db.session.add(product)
        db.session.commit()
        return redirect(url_for('index'))
    products = Product.query.all()
    return render_template('index.html', form=form, products=products)
```

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

```
@app.route('/category/<int:id>')
def category_products(id):
    category = Category.query.get(id)
    products = category.products
    return render_template('category.html', category=category, products=products)

if __name__ == '__main__':
    app.run(debug=True)
```

3. Сервис для сокращения заметок (аналог сокращения ссылок)

Создай приложение, которое позволяет сохранить длинный текст заметки и получить короткий код (например, 6 символов) для доступа к ней. Модель Note содержит поля: id, full_text (обязательное), short_code (уникальная строка длиной 6 символов), views (счётчик просмотров, по умолчанию 0), created_at (дата создания). Форма NoteForm содержит только поле full_text с валидацией. На главной странице (/) выведи форму и список всех заметок (короткий код, превью текста, количество просмотров). При отправке формы генерируется уникальный short_code (случайная строка из букв и цифр, проверка на уникальность). После сохранения перенаправление на /. По маршруту /<short_code> должен отображаться полный текст заметки и увеличиваться счётчик views. Ссылка на заметку должна вести на этот маршрут.

Входные данные:

Данные формы (текст заметки).

Выходные данные:

Список заметок, детальная страница заметки.

Пример:

После создания заметки с текстом «Привет, мир!» генерируется код AbCdEf. На главной появляется строка с этим кодом. При переходе по /AbCdEf отображается полный текст, счётчик увеличивается.

Решение:

```
import random, string
from flask import Flask, render_template, redirect, url_for
from flask_sqlalchemy import SQLAlchemy
from flask_wtf import FlaskForm
from wtforms import TextAreaField, SubmitField
from wtforms.validators import DataRequired
from datetime import datetime

app = Flask(__name__)
app.config['SECRET_KEY'] = 'secret'
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///notes.db'
db = SQLAlchemy(app)

class Note(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    full_text = db.Column(db.Text, nullable=False)
    short_code = db.Column(db.String(6), unique=True, nullable=False)
    views = db.Column(db.Integer, default=0)
    created_at = db.Column(db.DateTime, default=datetime.utcnow)
```

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

```
db.create_all()

def generate_short_code():
    chars = string.ascii_letters + string.digits
    while True:
        code = ''.join(random.choice(chars) for _ in range(6))
        if not Note.query.filter_by(short_code=code).first():
            return code

class NoteForm(FlaskForm):
    full_text = TextAreaField('Текст заметки', validators=[DataRequired()])
    submit = SubmitField('Создать заметку')

@app.route('/', methods=['GET', 'POST'])
def index():
    form = NoteForm()
    if form.validate_on_submit():
        code = generate_short_code()
        note = Note(full_text=form.full_text.data, short_code=code)
        db.session.add(note)
        db.session.commit()
        return redirect(url_for('index'))
    notes = Note.query.order_by(Note.created_at.desc()).all()
    return render_template('index.html', form=form, notes=notes)

@app.route('/<short_code>')
def view_note(short_code):
    note = Note.query.filter_by(short_code=short_code).first_or_404()
    note.views += 1
    db.session.commit()
    return render_template('note.html', note=note)

if __name__ == '__main__':
    app.run(debug=True)
```

4. Редактирование информации о книге

На основе модели Book из задания 1 создай маршрут /edit/<int:id>, который отображает форму, предзаполненную данными выбранной книги. Форма должна быть той же BookForm (или отдельной, но с теми же полями). При отправке POST-запроса данные обновляются в БД. После успешного обновления происходит перенаправление на главную страницу. Если книга не найдена – возвращать 404.

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

	ID книги в URL, данные формы.
	Выходные данные:
	Страница с формой редактирования, затем перенаправление.
	Пример:
	При переходе на /edit/1 отображается форма с текущими значениями книги «Война и мир». После изменения и отправки данные обновляются.
	Решение (дополнить предыдущее приложение):
	# Добавить в файл с моделями и формами
	@app.route('/edit/<int:id>', methods=['GET', 'POST'])
	def edit_book(id):
	book = Book.query.get_or_404(id)
	form = BookForm(obj=book) # предзаполнение
	if form.validate_on_submit():
	book.title = form.title.data
	book.author = form.author.data
	book.year = form.year.data
	book.rating = form.rating.data
	db.session.commit()
	return redirect(url_for('index'))
	return render_template('edit.html', form=form, book=book)
	5. Удаление книги с подтверждением
	Добавь маршрут /delete/<int:id>, который удаляет книгу из базы данных. Для безопасности реализуй подтверждение через отдельную страницу или через POST-запрос. Предложи простой вариант: при GET-запросе показывается страница с вопросом «Вы уверены?» и кнопками «Да» и «Нет»; при POST-запросе (нажатие «Да») – удаление и перенаправление на главную. Либо можно сделать удаление через кнопку на главной с отправкой POST-запроса (например, через форму с методом POST). В данном задании реализуй удаление через отдельный маршрут с подтверждением.
	Входные данные:
	ID книги в URL.
	Выходные данные:
	Страница подтверждения или перенаправление.
	Пример:
	При переходе на /delete/1 открывается страница с вопросом об удалении. При подтверждении книга исчезает из списка.
	Решение (дополнить):
	# Добавить маршрут
	@app.route('/delete/<int:id>', methods=['GET', 'POST'])
	def delete_book(id):
	book = Book.query.get_or_404(id)
	if request.method == 'POST':
	db.session.delete(book)
	db.session.commit()
	return redirect(url_for('index'))
	return render_template('confirm_delete.html', book=book)

	(Шаблон confirm_delete.html содержит форму с кнопкой подтверждения.)
Модуль 4, Фреймворк Flask, Проект	
Количество академических часов	
Формы контроля	
Диагностические инструменты	
Показатели и критерии оценивания	
Модуль 4, Фреймворк Flask, Структура проекта	
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 учетом 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 Марцинив Виктория Вадимовна
Документ подписан квалифицированной электронной подписью 04.08.2026
серийный номер 01ba3191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Варианты заданий	1. Создание модульной структуры проекта
	Создай структуру Flask-приложения с разделением на пакет app, содержащий модули: __init__.py, models.py, forms.py, views.py, а также папки templates и static. В корне проекта размести файлы: config.py, .env, .gitignore, requirements.txt, README.md. В config.py определи класс Config с параметрами SECRET_KEY и SQLALCHEMY_DATABASE_URI, которые должны загружаться из переменных окружения с помощью os.getenv(). В файле .env укажи тестовые значения этих переменных. В __init__.py создай фабрику create_app(), которая инициализирует Flask и SQLAlchemy. Напиши код для каждого файла.
	Входные данные:
	Нет (создание файлов).
	Выходные данные:
	Набор файлов с корректным кодом.
	Пример:
	Файл .env должен содержать:
	SECRET_KEY=mysecretkey
	DATABASE_URL=sqlite:///project.db
	Решение:
	config.py:
	python
	import os
	from dotenv import load_dotenv
	
	load_dotenv()
	
	class Config:
	SECRET_KEY = os.getenv('SECRET_KEY', 'dev-key')
	SQLALCHEMY_DATABASE_URI = os.getenv('DATABASE_URL', 'sqlite:///default.db')
	SQLALCHEMY_TRACK_MODIFICATIONS = False
	app/__init__.py:
	from flask import Flask
	from flask_sqlalchemy import SQLAlchemy
	from config import Config
	
	db = SQLAlchemy()
	
	def create_app():
	app = Flask(__name__)
	app.config.from_object(Config)
	db.init_app(app)
	return app
	2. Модель пользователя с хэшированием пароля
	Создай модель User в models.py с полями: id (первичный ключ), username (уникальная строка, обязательное), email (уникальная строка, обязательное), password_hash (строка, обязательное). Реализуй методы: set_password(password) – сохраняет хэш пароля (используй generate_password_hash), check_password(password) – проверяет пароль (используй check_password_hash). Модель должна наследовать UserMixin из flask_login. Добавь необходимые импорты и свяжи db с приложением (предполагается, что db импортирован из __init__.py).
	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

Нет (создание класса).
Выходные данные:
Класс User с методами.
Пример:
user = User(username='alice', email='alice@example.com')
user.set_password('qwerty123')
Решение:
app/models.py:

```
from flask_login import UserMixin
from werkzeug.security import generate_password_hash, check_password_hash
from app import db

class User(UserMixin, db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(80), unique=True, nullable=False)
    email = db.Column(db.String(120), unique=True, nullable=False)
    password_hash = db.Column(db.String(256), nullable=False)

    def set_password(self, password):
        self.password_hash = generate_password_hash(password)

    def check_password(self, password):
        return check_password_hash(self.password_hash, password)
```

3. Форма и маршрут для регистрации пользователя
Создай форму RegistrationForm в forms.py с полями: username, email, password, confirm_password (подтверждение пароля). Используй валидаторы: DataRequired, Email для email, EqualTo для confirm_password. Затем в views.py создай маршрут /register с методами GET и POST. При GET – отобрази шаблон register.html с формой. При POST – если форма валидна, создай пользователя, сохрани его в БД и перенаправь на страницу входа (маршрут /login). Не забудь добавить импорты redirect, url_for, render_template.
Входные данные:
Данные формы (POST-запрос).
Выходные данные:
Страница регистрации или перенаправление.
Пример:
После успешной отправки формы пользователь перенаправляется на /login.
Решение:
app/forms.py:

```
from flask_wtf import FlaskForm
from wtforms import StringField, PasswordField, SubmitField
from wtforms.validators import DataRequired, Email, EqualTo
```

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 RegistrationForm(FlaskForm):
    username = StringField('Имя пользователя', validators=[DataRequired()])
    email = StringField('Email', validators=[DataRequired(), Email()])
    password = PasswordField('Пароль', validators=[DataRequired()])
    confirm = PasswordField('Подтверждение', validators=[DataRequired(), EqualTo('password')])
    submit = SubmitField('Зарегистрироваться')
app/views.py:
from flask import render_template, redirect, url_for
from app import db
from app.forms import RegistrationForm
from app.models import User

@app.route('/register', methods=['GET', 'POST'])
def register():
    form = RegistrationForm()
    if form.validate_on_submit():
        user = User(username=form.username.data, email=form.email.data)
        user.set_password(form.password.data)
        db.session.add(user)
        db.session.commit()
        return redirect(url_for('login'))
    return render_template('register.html', form=form)
```

4. Форма и маршрут для входа (аутентификация)

Создай форму LoginForm в forms.py с полями: username, password, remember (BooleanField). В views.py добавь маршрут /login с методами GET и POST. При POST проверь, что пользователь существует и пароль верен (используй User.query.filter_by(username=...).first() и check_password()). Если всё верно – выполни login_user(user, remember=form.remember.data) и перенаправь на главную. В противном случае – выведи flash-сообщение об ошибке. Настрой LoginManager в __init__.py (задай login_view='login', login_message_category='warning'). Добавь декоратор @login_required к любому защищённому маршруту (например, /profile).

Входные данные:

Данные формы (POST-запрос).

Выходные данные:

Вход в систему или сообщение об ошибке.

Пример:

При успешном входе перенаправление на главную.

Решение:

app/forms.py (дополнить):

```
from wtforms import BooleanField

class LoginForm(FlaskForm):
    username = StringField('Имя пользователя', validators=[DataRequired()])
    password = PasswordField('Пароль', validators=[DataRequired()])
```

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

```
    remember = BooleanField('Запомнить меня')
    submit = SubmitField('Войти')
app/views.py (дополнить):
from flask_login import login_user, login_required, current_user
from flask import flash

@app.route('/login', methods=['GET', 'POST'])
def login():
    form = LoginForm()
    if form.validate_on_submit():
        user = User.query.filter_by(username=form.username.data).first()
        if user and user.check_password(form.password.data):
            login_user(user, remember=form.remember.data)
            return redirect(url_for('index'))
        flash('Неверное имя пользователя или пароль', 'danger')
    return render_template('login.html', form=form)

@app.route('/profile')
@login_required
def profile():
    return f'Привет, {current_user.username}!'
```

5. Добавление категории (только для авторизованных) с flash-сообщением

Создай модель Category в models.py с полями id и title. Создай форму CategoryForm с полем title и валидатором. Добавь маршрут /add_category, защищённый @login_required. При GET – отобрази форму, при POST – сохрани категорию в БД, отправь flash-сообщение «Категория успешно добавлена!» (категория success) и перенаправь на главную. В шаблоне (предполагается base.html) добавь отображение flash-сообщений из get_flashed_messages().

Входные данные:

Данные формы (POST-запрос).

Выходные данные:

Flash-сообщение и перенаправление.

Пример:

После добавления категории «IT-новости» появляется зелёное сообщение «Категория успешно добавлена!».

Решение:

```
app/models.py (дополнить):
class Category(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    title = db.Column(db.String(255), nullable=False)
app/forms.py (дополнить):
class CategoryForm(FlaskForm):
    title = StringField('Название категории', validators=[DataRequired()])
    submit = SubmitField('Добавить категорию')
app/views.py (дополнить):
from flask import flash
from app.models import Category
from app.forms import CategoryForm

@app.route('/add_category', methods=['GET', 'POST'])
@login_required
def add_category():
    form = CategoryForm()
    if form.validate_on_submit():
        category = Category(title=form.title.data)
        db.session.add(category)
        db.session.commit()
        flash('Категория успешно добавлена!', 'success')
        return redirect(url_for('index'))
    return render_template('add_category.html', form=form)
templates/base.html (фрагмент для вывода flash):
{% with messages = get_flashed_messages(with_categories=true) %}
  {% for category, message in messages %}
    <div class="alert alert{ {category } }">{ {message }}</div>
  {% endfor %}
{% endwith %}
```

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Марценя Виктория Вадимовна
сертификат номер 01ba0191c94f2dc94c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Модуль 4, Фреймворк Flask, 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.
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)
	Создай класс MusicAlbum с атрибутами: title (название), artist (исполнитель), year (год выпуска). Реализуй конструктор __init__ и метод info(), который выводит строку в формате:
	«{исполнитель} — {название} ({год})».
	В основной программе прочитай три строки (название, исполнитель, год), создай объект и вызови метод info().
	Входные данные:
	Три строки: название, исполнитель, год (целое число).
	Выходные данные:
	Одна строка.
	Пример ввода:
	The Dark Side of the Moon
	Pink Floyd
	1973
	Пример вывода:
	Pink Floyd — The Dark Side of the Moon (1973)
	Возможное решение:
	class MusicAlbum:
	  def __init__(self, title, artist, year):
	    self.title = title
	    self.artist = artist
	    self.year = year
	  def info(self):
	    print(f'{self.artist} — {self.title} ({self.year})')
	
	title = input()
	artist = input()
	year = int(input())
	album = MusicAlbum(title, artist, year)
	album.info()
	Задание 2 (написание кода)
	Создай класс Employee с атрибутами name (имя) и salary (зарплата – целое число). Реализуй:
	конструктор, который проверяет, что зарплата положительная; если нет – установи salary = 0 и выведи сообщение «Зарплата исправлена на 0»;
	метод bonus(percent), который увеличивает зарплату на указанный процент (число от 0 до 100);
	метод __str__, возвращающий строку «Сотрудник: {name}, зарплата: {salary}».
	В основной программе прочитай имя и зарплату через пробел. Создай объект, примени метод bonus с процентом 15, выведи строковое представление.
	Входные данные:
	Одна строка: имя и зарплата (целое число) через пробел.

Контур Крипто
Документ подписан квалифицированной электронной подписью 04.08.2026
владелец Марцена Виктория Вадимовна
серийный номер 01bab191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Выходные данные:
Сообщение об исправлении (если нужно) и строковое представление после бонуса.
Пример ввода:
Мария -2000
Пример вывода:
Зарплата исправлена на 0
Сотрудник: Мария, зарплата: 0
Возможное решение:

```
class Employee:
    def __init__(self, name, salary):
        self.name = name
        if salary > 0:
            self.salary = salary
        else:
            print('Зарплата исправлена на 0')
            self.salary = 0
    def bonus(self, percent):
        self.salary = int(self.salary * (1 + percent / 100))
    def __str__(self):
        return f'Сотрудник: {self.name}, зарплата: {self.salary}'


name, salary = input().split()
emp = Employee(name, int(salary))
emp.bonus(15)
print(emp)
```

Задание 3 (написание кода)
Создай класс BankAccount с приватными атрибутами __balance (баланс) и __owner (владелец). Реализуй:
конструктор, принимающий владельца и начальный баланс (по умолчанию 0);
геттеры balance и owner через @property;
сеттер balance, который не позволяет установить отрицательное значение (если пытаются – выводит «Баланс не может быть отрицательным» и не меняет);
методы deposit(amount) и withdraw(amount) для пополнения и снятия (с проверкой, что сумма положительна и не превышает баланс);
метод __str__, возвращающий «Владелец: {owner}, баланс: {balance}».
В основной программе прочитай владельца и начальный баланс через пробел. Затем прочитай сумму для пополнения и сумму для снятия (два целых числа).
Выполни операции и выведи строковое представление счёта.
Входные данные:
Первая строка – владелец и баланс; вторая строка – сумма пополнения; третья – сумма снятия.
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

Строковое представление счёта после операций.
Пример ввода:
Алексей 5000
1500
700
Пример вывода:
Владелец: Алексей, баланс: 5800
Возможное решение:

```
class BankAccount:
    def __init__(self, owner, balance=0):
        self.__owner = owner
        self.__balance = balance
    @property
    def balance(self):
        return self.__balance
    @balance.setter
    def balance(self, value):
        if value < 0:
            print('Баланс не может быть отрицательным')
        else:
            self.__balance = value
    @property
    def owner(self):
        return self.__owner
    @owner.setter
    def owner(self, value):
        self.__owner = value
    def deposit(self, amount):
        if amount > 0:
            self.__balance += amount
    def withdraw(self, amount):
        if 0 < amount <= self.__balance:
            self.__balance -= amount
    def __str__(self):
        return f'Владелец: {self.owner}, баланс: {self.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

```
owner, balance = input().split()
acc = BankAccount(owner, int(balance))
dep = int(input())
wid = int(input())
acc.deposit(dep)
acc.withdraw(wid)
print(acc)
```

Задание 4 (написание кода)

Создай класс Temperature с методами:

статический метод celsius_to_fahrenheit(c), который переводит градусы Цельсия в Фаренгейты по формуле F = C * 9/5 + 32;

метод класса from_fahrenheit(cls, f), который создаёт объект Temperature с градусами в Цельсиях, полученными из Фаренгейтов (C = (F - 32) * 5/9), округлёнными до 2 знаков;

метод __str__, возвращающий строку «{celsius}°C».

В основной программе прочитай одно вещественное число – градусы Цельсия. Выведи результат статического метода. Затем прочитай одно вещественное число – градусы Фаренгейта, создай объект через метод класса и выведи его.

Входные данные:

Два вещественных числа: сначала Цельсии, затем Фаренгейты.

Выходные данные:

Сначала значение в Фаренгейтах (статический метод), затем строковое представление объекта (в Цельсиях).

Пример ввода:

```
25
77
```

Пример вывода:

```
77.0
25.0°C
```

Возможное решение:

```
class Temperature:
    def __init__(self, celsius):
        self.celsius = round(celsius, 2)
    @staticmethod
    def celsius_to_fahrenheit(c):
        return c * 9/5 + 32
    @classmethod
    def from_fahrenheit(cls, f):
        c = (f - 32) * 5/9
        return cls(c)
    def __str__(self):
        return f'{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

```
c = float(input())
print(Temperature.celsius_to_fahrenheit(c))
f = float(input())
t = Temperature.from_fahrenheit(f)
print(t)
```

Задание 5 (написание кода)

Создай класс Fraction для работы с обыкновенными дробями. Конструктор принимает числитель и знаменатель, автоматически сокращает дробь (используй math.gcd). Реализуй методы:

__add__, __sub__, __mul__, __truediv__ – арифметические операции, возвращающие новый объект Fraction;

__str__ – возвращает строку «{num}/{den}».

В основной программе прочитай две дроби (каждая в виде двух целых чисел через пробел). Выведи результаты сложения, вычитания, умножения и деления этих дробей (каждый на новой строке).

Входные данные:

Две строки, каждая с двумя целыми числами (числитель и знаменатель).

Выходные данные:

Четыре строки – результаты операций.

Пример ввода:

2 5

3 4

Пример вывода:

23/20

-7/20

3/10

8/15

Возможное решение:

```
import math


class Fraction:
    def __init__(self, num, den):
        g = math.gcd(num, den)
        self.num = num // g
        self.den = den // g
    def __add__(self, other):
        num = self.num * other.den + other.num * self.den
        den = self.den * other.den
        return Fraction(num, den)
    def __sub__(self, other):
```

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

```
        num = self.num * other.den - other.num * self.den
        den = self.den * other.den
        return Fraction(num, den)
    def __mul__(self, other):
        return Fraction(self.num * other.num, self.den * other.den)
    def __truediv__(self, other):
        return Fraction(self.num * other.den, self.den * other.num)
    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)
print(f1 - f2)
print(f1 * f2)
print(f1 / f2)
```

Задание 6 (написание кода)

Создай класс Vector для представления вектора на плоскости с координатами x и y. Реализуй:

конструктор с параметрами x, y (по умолчанию 0);

метод __len__, возвращающий длину вектора (округлённую до целого);

методы сравнения __eq__, __lt__ (сравнение по длине);

методы __add__, __sub__ (сложение и вычитание векторов, возвращают новый объект);

метод __mul__ для умножения вектора на число (возвращает новый объект);

метод __str__ – возвращает строку «Vector({x}, {y})».

В основной программе прочитай четыре целых числа: x1, y1, x2, y2. Создай два вектора. Выведи их сумму, разность и результат сравнения по длине (True/False).

Входные данные:

Четыре целых числа (x1 y1 x2 y2) через пробел.

Выходные данные:

Три строки: сумма, разность, результат сравнения (True/False).

Пример ввода:

3 4 1 2

Пример вывода:

Vector(4, 6)

Vector(2, 2)

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

True
Возможное решение:

```
import math
from functools import total_ordering


@total_ordering
class Vector:
    def __init__(self, x=0, y=0):
        self.x = x
        self.y = y
    def __len__(self):
        return round(math.hypot(self.x, self.y))
    def __eq__(self, other):
        return math.hypot(self.x, self.y) == math.hypot(other.x, other.y)
    def __lt__(self, other):
        return math.hypot(self.x, self.y) < math.hypot(other.x, other.y)
    def __add__(self, other):
        return Vector(self.x + other.x, self.y + other.y)
    def __sub__(self, other):
        return Vector(self.x - other.x, self.y - other.y)
    def __mul__(self, scalar):
        return Vector(self.x * scalar, self.y * scalar)
    def __str__(self):
        return f'Vector({self.x}, {self.y})'


x1, y1, x2, y2 = map(int, input().split())
v1 = Vector(x1, y1)
v2 = Vector(x2, y2)
print(v1 + v2)
print(v1 - v2)
print(v1 > v2)   # используется __lt__ и total_ordering
```

Задание 7 (написание кода)
Создай класс Student с атрибутами name и course (курс от 1 до 4). Конструктор должен проверять корректность курса. Если курс не в диапазоне 1–4, выбрасывать собственное исключение InvalidCourseError с сообщением «Некорректный курс». Реализуй метод promote(), который переводит студента на следующий курс (если курс меньше 4), иначе выбрасывает исключение GraduateError с сообщением «Студент уже выпустился». Также реализуй метод __str__, возвращающий строку «Студент: {name}, курс {course}`.
В основной программе прочитай имя и курс через пробел. Попытайся создать объект и применить promote(). Если возникает исключение, выведи его сообщение. Если всё успешно, выведи строковое представление студента.
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: имя и курс (целое число).
Выходные данные:
Сообщение об ошибке (если есть) или строковое представление студента.
Пример ввода:
Иван 5
Пример вывода:
Некорректный курс
Возможное решение:

```
class InvalidCourseError(Exception):
    pass


class GraduateError(Exception):
    pass


class Student:
    def __init__(self, name, course):
        self.name = name
        if not (1 <= course <= 4):
            raise InvalidCourseError('Некорректный курс')
        self.course = course
    def promote(self):
        if self.course == 4:
            raise GraduateError('Студент уже выпустился')
        self.course += 1
    def __str__(self):
        return f'Студент: {self.name}, курс {self.course}'


try:
    name, course = input().split()
    s = Student(name, int(course))
    s.promote()
    print(s)
except (InvalidCourseError, GraduateError) 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

Задание 8 (написание кода)
Напиши декоратор @timer, который замеряет время выполнения функции (в секундах) и выводит его на экран. Используй модуль time. Декоратор должен работать для функций с любым количеством аргументов.
В основной программе определи функцию slow_function, которая вычисляет сумму чисел от 1 до n (n передаётся как аргумент) и возвращает её. Примени декоратор @timer к этой функции. Считай с клавиатуры целое число n, вызови функцию и выведи результат.
Входные данные:
Одно целое число n.
Выходные данные:
Сначала время выполнения (например, «Время: 0.0012 сек.»), затем результат суммы.
Пример ввода:
100000
Пример вывода:
Время: 0.0034 сек.
5000050000
Возможное решение:
:
```
import time


def timer(func):
    def wrapper(*args, **kwargs):
        start = time.time()
        result = func(*args, **kwargs)
        elapsed = time.time() - start
        print(f'Время: {elapsed:.4f} сек.')
        return result
    return wrapper


@timer
def slow_function(n):
    return sum(range(1, n + 1))


n = int(input())
print(slow_function(n))
```
Задание 9 (исправление ошибок)
Дана программа, которая должна создавать класс Book с конструктором и методом info(). Она содержит ошибки.
python
```
class Book:
```

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__(title, author):
        self.title = title
        self.author = author
    def info()
        print(f'{self.author} - {self.title}')


b = Book('1984', 'Оруэлл')
b.info()
```

Найди и исправь ошибки.

Входные данные:

Нет (программа не считывает ввод).

Выходные данные:

Ожидаемый вывод: «Оруэлл - 1984».

Возможное решение:

```
class Book:
    def __init__(self, title, author):   # добавили self
        self.title = title
        self.author = author
    def info(self):                    # добавили self и двоеточие
        print(f'{self.author} - {self.title}')


b = Book('1984', 'Оруэлл')
b.info()
```

Задание 10 (исправление ошибок)

Дана программа, использующая наследование, но метод super() вызван неправильно. Исправь ошибку.

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


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

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 = Student('Анна', 20, 3)
print(s.name, s.age, s.course)
```

Входные данные:

Нет.

Выходные данные:

«Анна 20 3».

Возможное решение:

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


class Student(Person):
    def __init__(self, name, age, course):
        super().__init__(name, age)   # исправлено: передача аргументов в скобках
        self.course = course


s = Student('Анна', 20, 3)
print(s.name, s.age, s.course)
```

Задание 11 (исправление ошибок)

Дана программа, реализующая инкапсуляцию с помощью @property, но сеттер написан с ошибкой.

```
class Temperature:
    def __init__(self, celsius):
        self.__celsius = celsius
    @property
    def celsius(self):
        return self.__celsius
    @celsius.setter
    def celsius(self, value):
        if value < -273:
            print('Температура ниже абсолютного нуля!')
        else:
            __celsius = value
```

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

```
t = Temperature(25)
t.celsius = -300
print(t.celsius)
```

Найди и исправь ошибку.

Входные данные:

Нет.

Выходные данные:

Сообщение «Температура ниже абсолютного нуля!» и затем 25.

Возможное решение:

```
class Temperature:
    def __init__(self, celsius):
        self.__celsius = celsius
    @property
    def celsius(self):
        return self.__celsius
    @celsius.setter
    def celsius(self, value):
        if value < -273:
            print('Температура ниже абсолютного нуля!')
        else:
            self.__celsius = value   # исправлено: self.__celsius


t = Temperature(25)
t.celsius = -300
print(t.celsius)
```

Задание 12 (исправление ошибок)

Дана программа с собственным исключением, но оно не перехватывается корректно. Исправь.

```
class MyError(Exception):
    pass


def divide(a, b):
    if b == 0:
        raise MyError('Деление на 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

	return a / b
	
	try:
	print(divide(10, 0))
	except Exception as e:
	print('Ошибка')
	Ожидается, что при делении на ноль будет выведено сообщение «Деление на ноль», а не просто «Ошибка». Исправь программу, чтобы выводилось сообщение из исключения.
	Входные данные:
	Нет.
	Выходные данные:
	«Деление на ноль».
	Возможное решение:
	class MyError(Exception):
	pass
	
	def divide(a, b):
	if b == 0:
	raise MyError('Деление на ноль')
	return a / b
	
	try:
	print(divide(10, 0))
	except MyError as e: # перехватываем конкретное исключение
	print(e)
	Задание 13 (краткий ответ)
	Вопрос: Какой декоратор используется для создания метода, который может работать с атрибутами класса (принимает cls вместо self)? Запиши название декоратора.
	Ответ: @classmethod
	Задание 14 (краткий ответ)
	Вопрос: Какой специальный метод вызывается при использовании функции print() для объекта и должен возвращать строку, понятную человеку? Запиши название метода (включая двойные подчёркивания).
	Ответ: str
	Задание 15 (краткий ответ)
	Вопрос: Какую встроенную функцию следует использовать в конструкторе дочернего класса, чтобы вызвать конструктор родительского класса? Запиши название функции.
	Ответ: super

Шкала оценивания, нижнее значение	0
Шкала оценивания, верхнее значение	15
Шкала оценивания, минимальный проходной балл	9
Базы данных, 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 01ba2191c94f2cb04c6f27cc019aedad
срок действия 05.12.2025 - 05.03.2027

Варианты заданий	Задание 1 (написание кода)
	Реализуй класс Stack (стек) с методами push(item) – добавляет элемент на вершину, pop() – удаляет и возвращает верхний элемент (если стек пуст, выводит «error» и возвращает None), size() – возвращает количество элементов. Используй список для хранения данных. В основной программе прочитай количество команд n, затем n строк команд: push X, pop, size. Для команд pop и size выведи результат (каждый на новой строке). Гарантируется, что pop не вызывается на пустом стеке (кроме проверки – тогда должен выводить «error»).
	Входные данные:
	Первая строка – целое число n. Далее n строк команд.
	Выходные данные:
	Результаты выполнения команд pop и size.
	Пример ввода:
	5
	push 10
	push 20
	size
	pop
	pop
	Пример вывода:
	2
	20
	10
	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):
	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]))

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

Задание 2 (написание кода)

Напиши программу, которая вычисляет значение арифметического выражения, записанного в обратной польской записи (ОПЗ). Выражение содержит целые числа и операторы +, -, *, / (целочисленное деление с округлением в сторону нуля, как в Python // для положительных, а для отрицательных используй int(a / b)). Используй стек. Входная строка содержит числа и операторы, разделённые пробелами. Выведи результат.

Входные данные:

Одна строка – выражение в ОПЗ.

Выходные данные:

Одно целое число – результат.

Пример ввода:

12 3 / 2 + 4 *

Пример вывода:

24

Возможное решение:

```
class Stack:
    def __init__(self):
        self.data = []
    def push(self, val):
        self.data.append(val)
    def pop(self):
        return self.data.pop()

tokens = input().split()
stack = Stack()
for t in tokens:
    if t in '+-*/':
        b = stack.pop()
        a = stack.pop()
        if t == '+':
            stack.push(a + b)
        elif t == '-':
            stack.push(a - b)
        elif t == '*':
            stack.push(a * b)
        else:  # '/'
            stack.push(int(a / b))
```

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:
        stack.push(int(t))
print(stack.pop())
```

Задание 3 (написание кода)

Создай класс Queue (очередь) на основе односвязного списка. Реализуй методы: push(item) – добавляет элемент в конец (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 аналогично заданию 1 и выведи результаты для pop и size.

Входные данные:

Первая строка – n, затем n команд: push X, pop, size.

Выходные данные:

Результаты pop и size.

Пример ввода:

```
6
push 1
push 2
push 3
size
pop
pop
```

Пример вывода:

```
3
1
2
```

Возможное решение:

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

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:
```

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

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

Задание 4 (написание кода)

Напиши программу с использованием PyQt6, которая создаёт окно с заголовком «Мой калькулятор». В окне размести два поля ввода (QLineEdit) для целых чисел и кнопку «Сумма». При нажатии на кнопку программа должна вычислять сумму чисел из полей и выводить результат в третьем поле (QLineEdit), которое должно быть только для чтения. Используй менеджер макетов QVBoxLayout. Окно должно иметь размеры 300×150.

Входные данные:

Нет (ввод осуществляется через GUI).

Выходные данные:

GUI-окно с функционалом.

Пример:

При вводе 5 и 8 в соответствующие поля и нажатии кнопки в поле результата появляется 13.

Возможное решение:

```
import sys
from PyQt6.QtWidgets import QApplication, QWidget, QVBoxLayout, QLineEdit, QPushButton
```

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 Calculator(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle('Мой калькулятор')
        self.setGeometry(100, 100, 300, 150)
        layout = QVBoxLayout()
        self.edit1 = QLineEdit()
        self.edit2 = QLineEdit()
        self.btn = QPushButton('Сумма')
        self.result = QLineEdit()
        self.result.setReadOnly(True)
        layout.addWidget(self.edit1)
        layout.addWidget(self.edit2)
        layout.addWidget(self.btn)
        layout.addWidget(self.result)
        self.setLayout(layout)
        self.btn.clicked.connect(self.calc)
    def calc(self):
        try:
            a = int(self.edit1.text())
            b = int(self.edit2.text())
            self.result.setText(str(a + b))
        except ValueError:
            self.result.setText('Ошибка')

if __name__ == '__main__':
    app = QApplication(sys.argv)
    w = Calculator()
    w.show()
    sys.exit(app.exec())
```

Задание 5 (написание кода)

Создай приложение PyQt6 с главным окном (QMainWindow). Добавь меню «Файл» с пунктом «Выход», который закрывает окно. В центральной части окна размести метку (QLabel) с текстом «Главное окно». Также добавь строку состояния (QStatusBar), которая при запуске показывает сообщение «Готово».

Входные данные:

Нет.

Выходные данные:

GUI-окно.

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:

```
import sys
from PyQt6.QtWidgets import QApplication, QMainWindow, QLabel

class MainWindow(QMainWindow):
    def __init__(self):
        super().__init__()
        self.setWindowTitle('Главное окно')
        self.setCentralWidget(QLabel('Главное окно'))
        menu = self.menuBar()
        file_menu = menu.addMenu('Файл')
        file_menu.addAction('Выход', self.close)
        self.statusBar().showMessage('Готово')

if __name__ == '__main__':
    app = QApplication(sys.argv)
    w = MainWindow()
    w.show()
    sys.exit(app.exec())
```

Задание 6 (написание кода)

Напиши SQL-запрос, который выводит названия фильмов и имена их режиссёров для фильмов, выпущенных после 2010 года. Используй таблицы movies (поля: id, title, year, duration, director_id) и directors (id, name). Отсортируй результат по году выпуска по убыванию. Запрос должен быть написан как строка для выполнения в Python.

Входные данные:

Нет (запрос выполняется к базе данных).

Выходные данные:

Таблица с колонками title и name.

Пример:

При наличии фильмов 2012 и 2015 годов запрос выведет их.

Возможное решение:

```
SELECT movies.title, directors.name
FROM movies
JOIN directors ON movies.director_id = directors.id
WHERE movies.year > 2010
ORDER BY movies.year DESC;
```

Задание 7 (написание кода)

Напиши программу на Python с использованием модуля sqlite3, которая подключается к базе данных library.db, создаёт таблицу books с полями id INTEGER PRIMARY KEY, title TEXT, author TEXT, year INTEGER, и добавляет в неё книгу «Мастер и Маргарита» (автор «Булгаков», год 1967). После выполнения программа должна закрыть соединение.

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:
В базе данных появляется таблица и запись.
Возможное решение:

```
import sqlite3
con = sqlite3.connect('library.db')
con.execute('''CREATE TABLE IF NOT EXISTS books
               (id INTEGER PRIMARY KEY, title TEXT, author TEXT, year INTEGER)''')
con.execute('INSERT INTO books(title, author, year) VALUES (?, ?, ?)',
            ('Мастер и Маргарита', 'Булгаков', 1967))
con.commit()
con.close()
```

Задание 8 (написание кода)
Реализуй сортировку вставками по возрастанию для списка целых чисел. Программа должна выводить состояние списка после каждой вставки (т.е. после того, как текущий элемент вставлен на своё место). Ввод: сначала целое число n – размер списка, затем n целых чисел через пробел. Выведи n-1 строку (для каждого шага, начиная со второго элемента). Если список уже отсортирован, всё равно выводи состояния для каждого шага.
Входные данные:
Первая строка – n, вторая – n целых чисел.
Выходные данные:
n-1 строк, каждая содержит элементы списка через пробел после очередной 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 9 (исправление ошибок)

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

Дана программа, которая должна реализовывать стек с методом get_max() для получения максимального элемента за O(1). Однако метод get_max() работает неправильно. Найди и исправь ошибку.

```
class StackMax:
    def __init__(self):
        self.items = []
        self.maxes = []
    def push(self, x):
        self.items.append(x)
        if not self.maxes or x > self.maxes[-1]:
            self.maxes.append(x)
    def pop(self):
        if self.items:
            if self.items[-1] == self.maxes[-1]:
                self.maxes.pop()
            return self.items.pop()
    def get_max(self):
        return self.maxes[-1] if self.maxes else None
```

Программа должна корректно обрабатывать случай, когда добавляются равные максимальные элементы (например, push 5, push 5 – оба должны сохраняться в maxes, чтобы при удалении одного из них максимум не потерялся). Исправь метод push.

Возможное решение:

```
class StackMax:
    def __init__(self):
        self.items = []
        self.maxes = []
    def push(self, x):
        self.items.append(x)
        if not self.maxes or x >= self.maxes[-1]:  # исправлено: >= вместо >
            self.maxes.append(x)
    def pop(self):
        if self.items:
            if self.items[-1] == self.maxes[-1]:
                self.maxes.pop()
            return self.items.pop()
    def get_max(self):
        return self.maxes[-1] if self.maxes else None
```

Задание 10 (исправление ошибок)

Дана программа на PyQt6, которая должна создавать окно с кнопкой и при нажатии выводить сообщение. Код содержит ошибку подключения сигнала. Найди и исправь.

```
import sys
from PyQt6.QtWidgets import QApplication, QWidget, QPushButton, QMessageBox
```

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 Window(QWidget):
    def __init__(self):
        super().__init__()
        btn = QPushButton('Нажми', self)
        btn.clicked.connect(show_message)  # ошибка: show_message не определена как метод
    def show_message(self):
        QMessageBox.information(self, 'Сообщение', 'Привет!')

if __name__ == '__main__':
    app = QApplication(sys.argv)
    w = Window()
    w.show()
    sys.exit(app.exec())
```

Исправь, чтобы при нажатии кнопки появлялось сообщение.

Возможное решение:

```
import sys
from PyQt6.QtWidgets import QApplication, QWidget, QPushButton, QMessageBox

class Window(QWidget):
    def __init__(self):
        super().__init__()
        btn = QPushButton('Нажми', self)
        btn.clicked.connect(self.show_message)  # исправлено: добавили self.
    def show_message(self):
        QMessageBox.information(self, 'Сообщение', 'Привет!')

if __name__ == '__main__':
    app = QApplication(sys.argv)
    w = Window()
    w.show()
    sys.exit(app.exec())
```

Задание 11 (исправление ошибок)

Дана программа, которая должна выводить количество фильмов каждого режиссёра, используя SQL-запрос. Код содержит синтаксическую ошибку в SQL. Найди и исправь.

```
import sqlite3
con = sqlite3.connect('movies.db')
cur = con.cursor()
```

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

```
cur.execute('''SELECT directors.name, COUNT(movies.id) AS film_count
               FROM directors
               LEFT JOIN movies ON directors.id = movies.director_id
               GROUP BY directors.id
               ORDER BY film_count DESC''')
for row in cur.fetchall():
    print(row)
con.close()
```

Возможное решение: (исправленный запрос)

```
cur.execute('''SELECT directors.name, COUNT(movies.id) AS film_count
               FROM directors
               LEFT JOIN movies ON directors.id = movies.director_id
               GROUP BY directors.id
               ORDER BY film_count DESC''')
```

Задание 12 (исправление ошибок)

Дана программа, реализующая связный список и поиск элемента по значению. Программа должна выводить позицию элемента (с 1) или -1, если не найден. Однако функция поиска содержит ошибку: она не обновляет позицию корректно. Найди и исправь.

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

# Создаём список: 10 -> 20 -> 30 -> None
head = Node(10, Node(20, Node(30)))

def find(head, target):
    pos = 0
    current = head
    while current:
        pos += 1
        if current.value == target:
            return pos
        current = current.next
    return -1

print(find(head, 20))  # должно вывести 2, но выводит? всё верно. Но если ошибка в том, что pos начинается с 0, а первый элемент должен быть 1. В условии сказано: нумерация с 1. В коде pos=0, затем pos+=1, так что верно. Придумаем другую ошибку: например, забыли увеличить pos. Тогда исправление: добавить pos += 1. Я добавлю такую ошибку.
```

Упростим: в коде забыли увеличить pos внутри цикла. Исправление: добавить pos += 1 перед проверкой или после.

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, next_node=None):
	  self.value = value
	  self.next = next_node
	
	head = Node(10, Node(20, Node(30)))
	
	def find(head, target):
	  pos = 1
	  current = head
	  while current:
	  if current.value == target:
	  return pos
	  current = current.next
	  pos += 1
	  return -1
	
	print(find(head, 20)) # 2
	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: Какой метод объекта Connection в библиотеке sqlite3 необходимо вызвать после выполнения INSERT, UPDATE или DELETE, чтобы изменения были сохранены в базе данных? Запиши название метода.
	Ответ: commit
	Задание 15 (краткий ответ)
	Вопрос: Какой класс в PyQt6 используется для создания кнопки? Запиши название класса.
	Ответ: QPushButton

Шкала оценивания, нижнее значение	0
Шкала оценивания, верхнее значение	15
Шкала оценивания, минимальный проходной балл	9
WEB и API, 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)
	В файле data.txt хранятся целые числа, разделённые пробелами (в нескольких строках). Напиши программу, которая вычисляет произведение всех чисел из файла. Если чисел нет, программа должна вывести 1. Используй менеджер контекста with.
	Входные данные:
	Файл data.txt с числами.
	Выходные данные:
	Одно целое число – произведение.
	Пример ввода (файл data.txt):
	2 3 4
	5 -2
	Пример вывода:
	-240
	(2*3*4*5*(-2) = -240)
	Возможное решение:
	prod = 1
	with open('data.txt', encoding='utf-8') as f:
	  for line in f:
	  for num in line.split():
	  prod *= int(num)
	print(prod)
	Задание 2 (написание кода)
	Скачай файл weather.json (структура аналогична примеру из материалов: список прогнозов с полем main.temp_min). Напиши программу, которая находит минимальную температуру (temp_min) за весь период и выводит её. Если файл не содержит данных, выведи 0.
	Входные данные:
	Файл weather.json.
	Выходные данные:
	Одно вещественное число – минимальная температура.
	Пример структуры:
	{
	  "list": [
	  {"main": {"temp_min": 5.2, "temp_max": 7.8}},
	  {"main": {"temp_min": -1.3, "temp_max": 2.5}}
	  ]
	}
	Пример вывода:
	-1.3
	Возможное решение:
	import json
	with open('weather.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)
if not data['list']:
    print(0)
else:
    min_temp = min(entry['main']['temp_min'] for entry in data['list'])
    print(min_temp)
```

Задание 3 (написание кода)

В файле movies.csv в формате CSV есть столбцы Title, Rating (вещественное число). Напиши программу, которая находит фильм с самым низким рейтингом (среди тех, у кого рейтинг указан). Если таких фильмов несколько, выведи первый по порядку в файле. Выведи строку в формате:
«Название: {название}, рейтинг: {рейтинг}».

Входные данные:

Файл movies.csv.

Выходные данные:

Одна строка.

Пример ввода (файл):

```
Title,Rating
The Shawshank Redemption,9.3
The Godfather,9.2
The Room,3.6
```

Пример вывода:

```
Название: The Room, рейтинг: 3.6
```

Возможное решение:

```
import csv
min_rating = float('inf')
min_title = ''
with open('movies.csv', newline='', encoding='utf-8') as f:
    reader = csv.DictReader(f)
    for row in reader:
        if row['Rating']:
            r = float(row['Rating'])
            if r < min_rating:
                min_rating = r
                min_title = row['Title']
print(f'Название: {min_title}, рейтинг: {min_rating}')
```

Задание 4 (написание кода)

С клавиатуры (через sys.stdin) вводится произвольное количество строк, каждая строка содержит имя файла в формате «имя.расширение». Напиши программу, которая группирует файлы по расширению и для каждой группы выводит (в алфавитном порядке расширений):
имена файлов (полностью) в алфавитном порядке,
строку «Количество файлов {расширение}: {количество}».

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

Каждая группа отделяется пустой строкой. В конце вывода пустая строка не требуется.
Входные данные:
Строки вида «имя.расширение» (конец ввода – Ctrl+D).
Выходные данные:
См. формат в примере.
Пример ввода:
a.txt
b.jpg
c.txt
d.png
Пример вывода:
b.jpg
Количество файлов jpg: 1

a.txt
c.txt
Количество файлов txt: 2

d.png
Количество файлов png: 1
Возможное решение:

```
import sys
groups = {}
for line in sys.stdin:
    name, ext = line.strip().split('.')
    groups.setdefault(ext, []).append(name + '.' + ext)
for ext in sorted(groups):
    for fname in sorted(groups[ext]):
        print(fname)
    print(f'Количество файлов {ext}: {len(groups[ext])}')
    print()
```

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).

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:
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_advanced.py, которая принимает два целых числа (позиционные аргументы) и опциональный ключ -o (или --op) с возможными значениями: add, sub, mul, pow. Значение по умолчанию – add. Программа должна вывести результат операции. В случае ошибки (например, возведение в степень с отрицательным показателем – обрабатывать как ошибку) вывести строку «Error». Используй модуль argparse.
Примеры вызова:

```
$ python calc_advanced.py 5 3
8
$ python calc_advanced.py 5 3 -o mul
15
$ python calc_advanced.py 2 5 -o pow
32
$ python calc_advanced.py 2 -1 -o pow
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,
    'pow': 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')
```

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

```
args = parser.parse_args()
try:
    print(ops[args.op](args.a, args.b))
except Exception:
    print('Error')
```

Задание 7 (написание кода)

Напиши программу, которая парсит страницу https://peps.python.org/ и подсчитывает количество PEP в секции с идентификатором accepted-peps (тег <section id="accepted-peps">). Внутри этой секции все PEP перечислены в таблице <table>. Подсчитай количество строк <tr> в этой таблице (включая заголовок? В условии уточним: нужно подсчитать все строки таблицы, включая заголовок? В оригинале считали все строки, но можно уточнить. В нашем случае будем считать все строки, включая заголовок. Если нужны только данные, нужно исключить заголовок, но в условии этого нет, поэтому считаем все.) Выведи полученное число.

Входные данные:

Нет.

Выходные данные:

Одно целое число.

Пример вывода:

10

Возможное 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='accepted-peps')
table = section.find('table')
rows = table.find_all('tr')
print(len(rows))
```

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:

Понедельник

Возможное решение:

```
import datetime
date_str = input()
date = datetime.datetime.strptime(date_str, '%d.%m.%Y')
```

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

```
weekdays = ['Понедельник', 'Вторник', 'Среда', 'Четверг', 'Пятница', 'Суббота', 'Воскресенье']
print(weekdays[date.weekday()])
```

Задание 9 (исправление ошибок)

Дана программа, которая должна проверять, встречается ли число в списке, но содержит ошибку (не рассматривает последний элемент). Исправь её.

```
numbers = [3, 5, 7, 9, 11, 13]
found = False
x = int(input())
for i in range(len(numbers) - 1):
    if numbers[i] == x:
        found = True
if found:
    print('Число найдено')
else:
    print('Число не найдено')
```

Входные данные:

Одно целое число x.

Выходные данные:

«Число найдено» или «Число не найдено».

Пример ввода:

```
13
```

Пример вывода:

```
Число найдено
```

Возможное решение:

```
numbers = [3, 5, 7, 9, 11, 13]
found = False
x = int(input())
for i in range(len(numbers)):   # исправлено: len(numbers) вместо len(numbers)-1
    if numbers[i] == x:
        found = True
        break                   # добавлен break
if found:
    print('Число найдено')
else:
    print('Число не найдено')
```

Задание 10 (исправление ошибок)

Программа должна читать JSON-файл и выводить значение ключа value. В коде ошибка – используется неправильный метод чтения. Исправь.

```
import json
```

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

```
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 11 (исправление ошибок)

Программа должна использовать argparse для получения двух чисел и выводить их сумму. Код содержит ошибку – аргументы не обрабатываются как целые числа. Исправь.

```
import argparse
parser = argparse.ArgumentParser()
parser.add_argument('a')
parser.add_argument('b')
args = parser.parse_args()
print(args.a + args.b)
```

Входные данные:

Аргументы командной строки.

Выходные данные:

Сумма чисел.

Пример вызова:

python prog.py 5 7 → вывод 12.

Возможное решение:

```
import argparse
parser = argparse.ArgumentParser()
parser.add_argument('a', type=int)   # добавлен type=int
parser.add_argument('b', type=int)   # добавлен type=int
args = parser.parse_args()
print(args.a + args.b)
```

Задание 12 (исправление ошибок)

Программа должна искать все даты в формате ДД.ММ.ГГГГ в тексте, но регулярное выражение написано с ошибкой (не учитывает однозначные числа). Исправь.

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 re
	text = input()
	pattern = r'\d{2}\.\d{2}\.\d{4}'
	dates = re.findall(pattern, text)
	print(' '.join(dates))
	Входные данные:
	Строка с датами.
	Пример ввода:
	4.10.1957 и 03.11.1957.
	Ожидаемый вывод:
	4.10.1957 03.11.1957.
	Возможное решение:
	import re
	text = input()
	pattern = r'\d{1,2}\.\d{1,2}\.\d{4}' # исправлено: {1,2} вместо {2}
	dates = re.findall(pattern, text)
	print(' '.join(dates))
	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
Фреймворк Flask, 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.
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 Flask-приложение с маршрутом /about, который возвращает HTML-страницу с заголовком <h1> «О нас» и абзацем «Этот сайт создан для обучения веб-разработке». Используй функцию render_template для отображения шаблона about.html. Шаблон должен находиться в папке templates. Для запуска приложения используй режим отладки.
	Входные данные:
	Нет.
	Выходные данные:
	HTML-страница с указанным содержимым.
	Пример:
	При переходе на /about отображается страница с заголовком и абзацем.
	Возможное решение:
	from flask import Flask, render_template
	
	app = Flask(__name__)
	
	@app.route('/about')
	def about():
	  return render_template('about.html')
	
	if __name__ == '__main__':
	  app.run(debug=True)
	Шаблон templates/about.html:
	<!DOCTYPE html>
	<html>
	<head><meta charset="utf-8"><title>О нас</title></head>
	<body>
	  <h1>О нас</h1>
	  Этот сайт создан для обучения веб-разработке.
	</body>
	</html>
	Задание 2 (написание кода)
	Напиши Flask-приложение с динамическим маршрутом /user/<string:username>, который отображает страницу с сообщением: «Привет, {username}!». Если пользователь ввел «admin», то вместо приветствия выведи «Добро пожаловать, администратор!». Используй только возврат строки с HTML-разметкой (без render_template).
	Входные данные:
	Имя пользователя в URL.
	Выходные данные:
	HTML-страница с приветствием.

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: /user/Anna → «Привет, Anna!»; /user/admin → «Добро пожаловать, администратор!».
Возможное решение:

```
from flask import Flask


app = Flask(__name__)


@app.route('/user/<string:username>')
def user(username):
    if username == 'admin':
        return '<h1>Добро пожаловать, администратор!</h1>'
    return f'<h1>Привет, {username}!</h1>'


if __name__ == '__main__':
    app.run(debug=True)
```

Задание 3 (написание кода)
Создай Flask-приложение с формой для добавления отзыва. Используй Flask-WTF. Форма должна содержать поля: name (StringField, обязательное), text (TextAreaField, обязательное), rating (SelectField, choices=[1,2,3,4,5]), submit (SubmitField). На главной странице (/) отобрази форму и список всех отзывов (глобальный список feedbacks). После отправки формы отзыв добавляется в список и происходит перенаправление на /. Используй CSRF-защиту.
Входные данные:
Данные формы.
Выходные данные:
Страница с формой и списком отзывов.
Пример:
После добавления отзыва «Отлично!» от пользователя «Анна» с рейтингом 5, он появляется в списке.
Возможное решение:

```
from flask import Flask, render_template, redirect, url_for
from flask_wtf import FlaskForm
from wtforms import StringField, TextAreaField, SelectField, SubmitField
from wtforms.validators import DataRequired


app = Flask(__name__)
app.config['SECRET_KEY'] = 'mysecret'


feedbacks = []
```

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 FeedbackForm(FlaskForm):
    name = StringField('Имя', validators=[DataRequired()])
    text = TextAreaField('Отзыв', validators=[DataRequired()])
    rating = SelectField('Оценка', choices=[(1, '1'), (2, '2'), (3, '3'), (4, '4'), (5, '5')])
    submit = SubmitField('Добавить')


@app.route('/', methods=['GET', 'POST'])
def index():
    form = FeedbackForm()
    if form.validate_on_submit():
        feedbacks.append({
            'name': form.name.data,
            'text': form.text.data,
            'rating': form.rating.data
        })
        return redirect(url_for('index'))
    return render_template('index.html', form=form, feedbacks=feedbacks)


if __name__ == '__main__':
    app.run(debug=True)
```

Задание 4 (написание кода)

Создай модель Post для SQLAlchemy с полями: id (первичный ключ), title (строка, уникальная, обязательная, до 100 символов), content (текст, обязательное), created_at (дата и время, по умолчанию текущее). Подключи базу данных SQLite с именем blog.db к Flask-приложению. Создай таблицы с помощью db.create_all(). Добавь одну запись в таблицу Post с названием «Первый пост» и содержимым «Это мой первый пост» в коде приложения после создания таблиц.

Входные данные:

Нет.

Выходные данные:

В базе данных появляется таблица и запись.

Возможное решение:

```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy
from datetime import datetime


app = Flask(__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

```
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///blog.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
db = SQLAlchemy(app)


class Post(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    title = db.Column(db.String(100), unique=True, nullable=False)
    content = db.Column(db.Text, nullable=False)
    created_at = db.Column(db.DateTime, default=datetime.utcnow)


with app.app_context():
    db.create_all()
    if not Post.query.first():
        post = Post(title='Первый пост', content='Это мой первый пост')
        db.session.add(post)
        db.session.commit()
```

Задание 5 (написание кода)

Напиши функцию представления для маршрута /posts, которая выводит все записи из модели Post (из задания 4) в виде HTML-списка (<ul>), где каждый элемент списка содержит заголовок поста и ссылку на его детальную страницу /post/<id>. Используй шаблон posts.html. В шаблоне передай список постов.

Входные данные:

Нет.

Выходные данные:

HTML-страница со списком постов.

Пример:

При переходе на /posts отображается список с заголовками и ссылками «Читать далее».

Возможное решение (фрагмент):

```
@app.route('/posts')
def posts():
    all_posts = Post.query.all()
    return render_template('posts.html', posts=all_posts)
```

Шаблон posts.html:

```
<ul>
{% for post in posts %}
    <li><a href="{{ url_for('post_detail', id=post.id) }}">{{ post.title }}</a></li>
{% endfor %}
</ul>
```

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 6 (написание кода)
Создай Flask-приложение с системой аутентификации. Модель User содержит поля id, username (уникальный, обязательный), password_hash (обязательный). Реализуй метод set_password(password) для хэширования пароля (используй werkzeug.security.generate_password_hash). Создай форму регистрации RegistrationForm с полями username, password, confirm (с валидацией EqualTo('password')). Маршрут /register обрабатывает GET и POST, при успешной регистрации создаёт пользователя и перенаправляет на /login.
Входные данные:
Данные формы регистрации.
Выходные данные:
Перенаправление на страницу входа.
Пример:
После заполнения формы и отправки пользователь перенаправляется на /login.
Возможное решение:

```
from flask import Flask, render_template, redirect, url_for
from flask_sqlalchemy import SQLAlchemy
from flask_wtf import FlaskForm
from wtforms import StringField, PasswordField, SubmitField
from wtforms.validators import DataRequired, EqualTo
from werkzeug.security import generate_password_hash


app = Flask(__name__)
app.config['SECRET_KEY'] = 'secret'
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///users.db'
db = SQLAlchemy(app)


class User(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(80), unique=True, nullable=False)
    password_hash = db.Column(db.String(256), nullable=False)
    def set_password(self, password):
        self.password_hash = generate_password_hash(password)


class RegistrationForm(FlaskForm):
    username = StringField('Логин', validators=[DataRequired()])
    password = PasswordField('Пароль', validators=[DataRequired()])
    confirm = PasswordField('Подтвердите пароль', validators=[DataRequired(), EqualTo('password')])
    submit = SubmitField('Зарегистрироваться')
```

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

```
@app.route('/register', methods=['GET', 'POST'])
def register():
    form = RegistrationForm()
    if form.validate_on_submit():
        user = User(username=form.username.data)
        user.set_password(form.password.data)
        db.session.add(user)
        db.session.commit()
        return redirect(url_for('login'))
    return render_template('register.html', form=form)
```

Задание 7 (написание кода)

Создай маршрут /profile, защищённый декоратором @login_required (из Flask-Login), который отображает страницу с сообщением «Привет, {current_user.username}!» и кнопкой «Выйти», которая перенаправляет на маршрут /logout. Реализуй функцию logout, которая завершает сессию и перенаправляет на главную. Используй LoginManager с настройкой login_view='login'.

Входные данные:

Нет (пользователь должен быть авторизован).

Выходные данные:

Страница профиля или перенаправление на вход.

Пример:

После входа пользователь видит своё имя и кнопку выхода.

Возможное решение:

```
from flask_login import LoginManager, login_required, current_user, logout_user, login_user


login_manager = LoginManager(app)
login_manager.login_view = 'login'


@login_manager.user_loader
def load_user(user_id):
    return User.query.get(int(user_id))


@app.route('/profile')
@login_required
def profile():
    return f'<h1>Привет, {current_user.username}!</h1><a href="/logout">Выйти</a>'
```

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

```
@app.route('/logout')
def logout():
    logout_user()
    return redirect('/')
```

Задание 8 (написание кода)

Выполни рефакторинг монолитного Flask-приложения: создай пакет app с модулями models.py, forms.py, views.py, а также __init__.py с фабрикой create_app. Вынеси настройки в config.py, а секретные ключи и URI базы данных загружай из переменных окружения через os.getenv() (используй python-dotenv). В корне проекта создай .env с переменными SECRET_KEY и DATABASE_URL. Напиши фабрику, которая инициализирует Flask, конфигурацию и SQLAlchemy.

Входные данные:

Нет (создание файлов).

Выходные данные:

Структура проекта с корректными импортами.

Возможное решение (фрагменты):

.env:

```
SECRET_KEY=mysecretkey
DATABASE_URL=sqlite:///project.db
```

config.py:

```
import os
from dotenv import load_dotenv
load_dotenv()


class Config:
    SECRET_KEY = os.getenv('SECRET_KEY')
    SQLALCHEMY_DATABASE_URI = os.getenv('DATABASE_URL')
    SQLALCHEMY_TRACK_MODIFICATIONS = False
```

app/__init__.py:

```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy
from config import Config


db = SQLAlchemy()


def create_app():
    app = Flask(__name__)
    app.config.from_object(Config)
```

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

```
    db.init_app(app)
    return app
```

Задание 9 (исправление ошибок)

Дана программа, которая должна отображать сумму двух чисел из динамического URL, но содержит ошибку (используется неправильный конвертер). Исправь.

```
from flask import Flask
app = Flask(__name__)


@app.route('/sum/<a>/<b>')
def sum(a, b):
    return f'Сумма: {a + b}'


if __name__ == '__main__':
    app.run()
```

Ошибка: a и b принимаются как строки, а нужно как числа. Исправь, чтобы при переходе на /sum/5/7 выводилось 12.

Возможное решение:

```
from flask import Flask
app = Flask(__name__)


@app.route('/sum/<int:a>/<int:b>')   # добавлены конвертеры int
def sum(a, b):
    return f'Сумма: {a + b}'


if __name__ == '__main__':
    app.run()
```

Задание 10 (исправление ошибок)

Программа должна добавлять отзывы в список, но форма не проходит валидацию из-за отсутствия CSRF-токена. Исправь.

```
from flask import Flask, render_template, redirect
from flask_wtf import FlaskForm
from wtforms import StringField, SubmitField


app = Flask(__name__)
# забыли SECRET_KEY
```

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 ReviewForm(FlaskForm):
    text = StringField('Отзыв')
    submit = SubmitField('Отправить')


reviews = []


@app.route('/', methods=['GET', 'POST'])
def index():
    form = ReviewForm()
    if form.validate_on_submit():
        reviews.append(form.text.data)
        return redirect('/')
    return render_template('index.html', form=form)
```

Исправь, добавив необходимую конфигурацию.

Возможное решение:

```
app.config['SECRET_KEY'] = 'any_secret_key'   # добавлено
```

Задание 11 (исправление ошибок)

Программа пытается изменить запись в базе данных, но изменения не сохраняются. Найди и исправь ошибку.

```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy
app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///test.db'
db = SQLAlchemy(app)


class Item(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(50))


with app.app_context():
    db.create_all()
    item = Item.query.first()
```

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 item:
	item.name = 'Новое имя'
	db.session.add(item) # здесь ошибка: после изменения add не нужен, но он не вредит
	# отсутствует commit
	Проблема: после изменения атрибута и добавления в сессию не вызывается commit(). Исправь, добавив db.session.commit().
	Возможное решение:
	db.session.commit() # добавлено после add
	Задание 12 (исправление ошибок)
	В приложении с аутентификацией декоратор @login_required не работает, потому что LoginManager не настроен. Исправь, добавив необходимую настройку.
	from flask import Flask
	from flask_login import LoginManager
	
	app = Flask(__name__)
	# нет SECRET_KEY и настройки login_manager
	
	@login_manager.user_loader
	def load_user(user_id):
	return User.query.get(int(user_id))
	Исправь, добавив SECRET_KEY и инициализацию LoginManager.
	Возможное решение:
	app.config['SECRET_KEY'] = 'secret'
	login_manager = LoginManager(app)
	login_manager.login_view = 'login'
	Задание 13 (краткий ответ)
	Вопрос: Какой декоратор в Flask используется для связывания функции с URL-адресом? Запиши название декоратора.
	Ответ: @app.route
	Задание 14 (краткий ответ)
	Вопрос: Какой метод объекта db.session нужно вызвать после добавления или изменения объекта модели, чтобы сохранить изменения в базе данных? Запиши название метода.
	Ответ: commit
	Задание 15 (краткий ответ)
	Вопрос: Как называется файл, который должен находиться в папке пакета Python, чтобы эта папка считалась пакетом и из неё можно было импортировать модули? Запиши имя файла (включая расширение).
	Ответ: init.py

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 Марщеня 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. 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.	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. Мигель Гринберг: Разработка веб-приложений с использованием Flask на языке Python. — ДМК-Пресс, 2016 г. — 272с.

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