

Государственное автономное нетиповое образовательное учреждение
Свердловской области «Дворец молодёжи»
Центр цифрового образования детей «IT-куб. Верхняя Пышма»
Центр цифрового образования «IT-куб»
Центр цифрового образования детей «IT-куб. Солнечный»
Центр цифрового образования детей «IT-куб. Арамиль»

Принята на заседании
научно-методического совета
ГАНОУ СО «Дворец молодёжи»
Протокол № 3 от 14.05.2026 г.

Утверждена Директором
ГАНОУ СО «Дворец молодёжи»
А. Н. Слизько
Приказ № 682-д от 14.05.2026 г.

Дополнительная общеобразовательная общеразвивающая программа
технической направленности

«Программирование на Python»
(стартовый, базовый, продвинутый уровни)

Возраст обучающихся: 12–17 лет
Объём общеразвивающей программы: 432 часа
Срок реализации: 3 года

СОГЛАСОВАНО:

Начальник центра цифрового образования
детей «IT-куб. Верхняя Пышма»
Е.Г. Евстафьева
Начальник центра цифрового образования
«IT-куб» А.А. Лаптева
Начальник центра цифрового образования
детей «IT-куб. Солнечный»
О.А. Чуенко
Начальник центра цифрового образования
детей «IT-куб. Арамиль»
В.А. Сырникова

Авторы-составители:
Дюкина В.Д., Иванов А.В.,
Сафин К.Р., Шмелев А.А.,
педагоги дополнительного
образования;
Кадникова Н.С.,
методист;
Ушенин В.П.,
педагог-организатор

I. Комплекс основных характеристик программы

1. Пояснительная записка

В настоящее время навыки программирования могут помочь людям не только освоить компьютер более эффективно, но и расширить свои возможности в различных областях, таких как наука, технологии, бизнес и другие. Начиная с основ программирования, человек может улучшить свою аналитическую способность, логику мышления и креативное мышление.

Одним из языков программирования, с которым рекомендуют знакомиться начинающим программистом — это Python. Данный выбор обусловлен тем, что синтаксис языка достаточно прост и интуитивно понятен тем, кто только начинает знакомиться с программированием в целом. После изучения Python знакомство с более сложными языками, например такими как C++ и C#, проходит более просто и эффективно, так как основные знания уже имеются. При этом данный язык программирования является очень востребованным. Он применяется в разных областях, начиная от веб-разработка и до обучения искусственного интеллекта.

Программа «Программирование на Python» (далее – Программа) предназначена для изучения программирования на языке Python. Программа является каркасной, что позволяет использовать в процессе обучения программное обеспечение в зависимости от материально-технических возможностей учебных аудиторий, а также знаний и навыков преподавателя.

Дополнительная общеобразовательная общеразвивающая программа «Программирование на Python» имеет техническую направленность, ориентирована на изучение различных аспектов алгоритмизации задач и разработки программ, синтаксических конструкций языка Python, применение языка программирования Python в различных областях IT-сферы.

Программа разработана с учетом требований следующих нормативных правовых актов и государственных программных документов:

- Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;

- Федеральный закон Российской Федерации от 14.07.2022 № 295-ФЗ «О внесении изменений в Федеральный закон «Об образовании в Российской Федерации»;
- Федеральный закон от 24.07.1998 № 124-ФЗ «Об основных гарантиях прав ребёнка в РФ»;
- Концепция развития дополнительного образования детей до 2030 года, утвержденная распоряжением Правительства Российской Федерации от 31 марта 2022 г. № 678-р;
- Указ Президента российской Федерации от 21.07.2020 № 474 «О национальных целях развития Российской Федерации на период до 2030 года»;
- Указ Президента российской Федерации от 09.11.2022 № 809 «Об утверждении Основ государственной политики по сохранению и укреплению традиционных российских духовно-нравственных ценностей»;
- Постановление Правительства Российской Федерации от 11.10.2023 № 1678 «Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;
- Приказ Министерства Просвещения РФ от 27.07.2022 г. № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Приказ Министерства Просвещения Российской Федерации от 03.09.2019 г. № 467 «Об утверждении Целевой модели развития региональных систем дополнительного образования детей»;
- Приказ Министерства труда и социальной защиты Российской Федерации от 22.09.2021 г. № 652н «Об утверждении профессионального стандарта «Педагог дополнительного образования детей и взрослых»;
- Приказ Министерства науки и высшего образования РФ и Министерства просвещения РФ от 5 августа 2020 г. № 882/391

«Об утверждении Порядка организации и осуществления образовательной деятельности при сетевой форме реализации образовательных программ»;

- Постановление Главного государственного санитарного врача РФ от 28 сентября 2020 г. № 28 «Об утверждении СанПиН 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;

- Постановление Главного государственного санитарного врача РФ от 28 января 2021 г. № 2 «Об утверждении санитарных правил и норм»;

- Письмо Министерства образования и науки России от 18.11.2015 г. № 09-3242 «О направлении информации» (вместе с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ»);

- Письмо Министерства образования и науки России от 28.08.2015 № АК-2563/05 «О методических рекомендациях» (вместе с «Методическими рекомендациями по организации образовательной деятельности с использованием сетевых форм реализации образовательных программ»);

- Приказ Министерства образования и молодежной политики Свердловской области от 29.06.2023 № 9 785-Д «Об утверждении Требований к условиям и порядку оказания государственной услуги в социальной сфере «Реализация дополнительных образовательных программ в соответствии с социальным сертификатом»;

- Положение о дополнительных общеобразовательных общеразвивающих программах ГАНОУ СО «Дворец молодёжи», утвержденное приказом от 14.05.2020 № 269-д.

- Положение о сетевой форме реализации образовательных программ в ГАНОУ СО «Дворец молодёжи», утвержденное приказом от 08.11.2021 № 947-д.

Актуальность программы обусловлена популярностью языка программирования Python в IT-сфере. Python — это язык программирования, который широко используется в интернет-приложениях, разработке

программного обеспечения, науке о данных и машинном обучении. Знакомство с данным языком позволяет подростку окунуться в мир IT-разработки и дает возможность рассмотреть разные его сферы.

Отличительной особенностью программы является то, что в рамках изучения программы обучающиеся постоянно будут сталкиваться с необходимостью самостоятельной работы над заданиями: обучающиеся учатся решать задачи без помощи преподавателя. Для этого в содержание курса включены задания, в которых для решения задачи необходимо найти информацию самостоятельно; может потребоваться устранение ошибки, которую не так просто обнаружить. Все эти знания, умения и практические навыки решения актуальных задач, полученные на занятиях, готовят обучающихся к самостоятельной проектно-исследовательской деятельности с применением современных технологий.

Программа обучения предусматривает освоение языка программирования Python с акцентом на практическое применение: полученные знания и навыки непосредственно применимы при разработке IT-решений под задачи реальных заказчиков.

Полученный объем знаний и сформированные компетенции позволят обучающимся успешно продолжить образовательный путь по программам «Мобильная разработка», «Основы программирования на языке Python. Лицей Академии Яндекса», «Основы промышленного программирования. Лицей Академии Яндекса», а также будут востребованы для дальнейшего обучения в профильных средних специальных и высших учебных заведениях.

Адресат программы. Программа предназначена для подростков в возрасте 12–17 лет.

Возрастные особенности группы

Содержание программы учитывает возрастные и психологические особенности подростков 12–14, 15–17 лет, которые определяют выбор форм проведения занятий с обучающимися.

12–14 лет – подростковый период. Референтно-значимый тип

деятельности: проектная деятельность (встреча замысла и результата как авторское действие подростка), проявление себя в общественно значимых ролях (выход в настоящую взрослую действительность). Характерными особенностями подросткового возраста есть стремление к самообразованию и самовоспитанию, полная определенность склонностей и профессиональных интересов. Характерная особенность – личное самосознание, сознательное проявление индивидуальности. Ведущая потребность – самоутверждение. В подростковый период стабилизируются интересы детей.

15–17 лет – юношеский возраст. Ведущая деятельность – учебно-профессиональная. Завершение физического и психического созревания. Социальная готовность к общественно полезному производительному труду. В отличие от подросткового возраста, где проявление индивидуальности осуществляется благодаря самоидентификации – «кто я», в юношеском возрасте индивидуальность выражается через самопроявление – «как я влияю». Также следует отметить, что подростки в возрасте 15–17 лет характеризуются такими психическими процессами, как изменение структуры личности и возникновение интереса к ней, развитие абстрактных форм мышления, становление более осознанного и целенаправленного характера деятельности, проявление стремления к самостоятельности и независимости, формирование самооценки. Эти процессы позволяют положить начало формированию начального профессионального самоопределения обучающихся.

Форма организации образовательной деятельности – групповая.

Место проведения занятий: ЦЦО «IT-куб» г. Екатеринбург, ул. Красных командиров 11 а, ЦЦОД «IT-куб г. Верхняя Пышма» г. Верхняя Пышма, пр. Успенский, 2Г, Центр цифрового образования детей «IT-куб г. Арамилы» г. Арамилы, ул. Щорса, 55, Центр цифрового образования детей «IT-куб. Солнечный» г. Екатеринбург, ул. Чемпионов, 11

Режим занятий: продолжительность одного академического часа - 45 минут. Перерыв между учебными занятиями – 10 минут. Общее количество

часов в неделю – 4 часа. Занятия проводятся 2 раза в неделю по 2 часа.

Срок освоения общеразвивающей программы определяется содержанием программы и составляет 3 года (144 часа в год).

Особенности организации образовательного процесса.

Дополнительная общеобразовательная общеразвивающая программа «Программирование на Python» состоит из трех модулей: первый модуль является стартовым уровнем сложности, второй – базовым, третий – продвинутым.

Формы обучения: очная, возможна реализация программы очно с применением электронного обучения и дистанционных образовательных технологий.

Возможна реализация дополнительной общеобразовательной общеразвивающей программы в сетевой форме. ГАНОУ СО «Дворец молодёжи» является базовой организацией, организация-участник определяется на основании заключенного договора о сетевой форме реализации программ

Объем общеразвивающей программы: 432 часа.

По уровню освоения программа является разноуровневой.

«Стартовый уровень» обеспечивает возможность обучения детей с любым уровнем подготовки.

Стартовый уровень образовательной программы рассчитан на обучающихся в возрасте 12–16 лет, проявляющих интерес к IT-технологиям и приобретению навыков программирования.

Зачисление детей на стартовый уровень производится без предварительного отбора (свободный набор).

К концу учебного года обучающиеся получают первичные навыки программирования, понятие о базовых конструкциях в Python; изучат основы функционального программирования.

«Базовый уровень» обеспечивает возможность обучения детей с первичными навыками программирования. Базовый уровень рассчитан на

обучающихся в возрасте 13–17 лет, демонстрирующих интерес к языкам программирования.

Зачисление обучающихся на базовый уровень обучения после завершения стартового уровня программы «Программирование на Python» или программы «Основы программирования на языке Python. Лицей Академии Яндекса» производится по результатам успешной сдачи итоговой аттестации (защиты итогового проекта), либо без прохождения стартового уровня после проведения тестирования.

Обучение по программе базового уровня нацелено на углубление и структурирование знаний основ современных языков программирования; умение на практике использовать сложные структуры данных; работать со специальными средствами и библиотеками языка Python.

К концу обучения обучающиеся способны самостоятельно определять профессиональные задачи и пути решения; писать грамотный код; находить и обрабатывать ошибки в коде; разрабатывать эффективные алгоритмы и программы на основе изученного языка программирования Python; способны самостоятельно изучать новые технологии.

«Продвинутый уровень» обеспечивает возможность углублённого обучения детей, обладающих прочными навыками программирования и опытом разработки программных решений. Продвинутый уровень рассчитан на обучающихся в возрасте 14–17 лет, проявляющих заинтересованность в углубленном освоении языков программирования и проектной деятельности.

Зачисление обучающихся на продвинутый уровень обучения производится после успешного завершения базового уровня программы «Программирование на Python» или программы «Основы промышленного программирования. Лицей Академии Яндекса» — по результатам итоговой аттестации (защиты итогового проекта); без прохождения предыдущих уровней — при условии успешного прохождения углублённого тестирования.

Обучение по программе продвинутого уровня нацелено на изучение принципов объектно-ориентированного и функционального

программирования в контексте реальных задач, понимание алгоритмов и структур данных повышенной сложности, знакомство с принципами проектирования масштабируемых и безопасных приложений.

К концу обучения обучающиеся способны выбирать оптимальные алгоритмы и структуры данных для решения задач различной сложности, оптимизировать производительность кода, выявлять и устранять узкие места, работать в команде над крупными проектами, используя современные инструменты совместной разработки, самостоятельно осваивать новые языки программирования, фреймворки и технологии, готовить и публично защищать технические проекты, аргументированно обосновывать принятые архитектурные и алгоритмические решения.

2. Цели и задачи программы

Цель программы: формирование у обучающихся научно-технических компетенций и практических навыков в области программирования на языке Python.

Для успешной реализации поставленной цели необходимо решить ряд обучающих, развивающих и воспитательных задач:

Обучающие:

- сформировать навыки программирования. познакомить с основными концепциями программирования на языке Python;
- сформировать навык эффективного решения задач на языке Python;
- обучить написанию эффективного и чистого кода для различных проблемных ситуаций;
- обучить основам применения языка программирования Python в различных областях IT-сферы;
- обучить основам создания MVP проектов на языке программирования Python.

Развивающие:

- развивать умения и навыки: планирование рабочей деятельности по реализации замысла, предвидение результата и его достижения, внесение корректировок в первоначальный замысел;
- формировать навыки работы с различными источниками информации, умение самостоятельно искать, извлекать и отбирать необходимую информацию;
- познакомить с правилами индивидуального и коллективного безопасного поведения при работе с компьютерной техникой, санитарно-гигиеническими нормами;
- формировать навык изложения мысли в четкой логической последовательности, отстаивания точки зрения, анализа ситуации и самостоятельного поиска ответов, путем логических рассуждений;

- развивать умение планирования создания продукта от идеи до действующего прототипа / макета, с учетом выстраивания междисциплинарных связей в области информатики, математики, физики, мехатроники.

Воспитательные:

- способствовать воспитанию этики групповой работы, отношений делового сотрудничества, взаимоуважения;
- способствовать развитию основ коммуникативных отношений внутри проектных групп и в коллективе в целом;
- способствовать воспитанию упорства в достижении результата;
- способствовать формированию целеустремлённости, организованности, ответственного отношения к труду и уважительного отношения к окружающим.

2.1. Цели и задачи I модуля (стартовый уровень)

Цель стартового уровня: получение навыков самостоятельного написания кода и разработки эффективных алгоритмов программирования на языке Python .

Для успешной реализации поставленной цели необходимо решить ряд педагогических, развивающих и воспитательных задач:

Обучающие:

- познакомить с базовыми понятиями и принципами функционального и объектно-ориентированного программирования;
- сформировать базовые навыки работы с основными конструкциями языка программирования Python;
- формировать навыки решения прикладных задач на языке Python;
- сформировать навыки программирования оборудования с помощью Python;
- сформировать навыки работы с информацией, необходимой для программирования на языке Python.

- усовершенствовать навыки работы с компьютером и офисными программами и/или обучить использованию прикладных программ для оформления проектов.

Развивающие:

- способствовать развитию интереса к программированию и техническим видам творчества;
- способствовать развитию самостоятельности и творческого подхода к решению задач;
- способствовать развитию умения формулировать, аргументировать и отстаивать своё мнение;
- способствовать развитию умения самостоятельно искать, извлекать и отбирать необходимую для решения учебных задач информацию;
- познакомить с правилами индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

Воспитательные:

- способствовать воспитанию этики групповой работы, отношению делового сотрудничества, взаимоуважения;
- способствовать воспитанию организованности, усидчивости и внимательности;
- способствовать воспитанию аккуратности при работе с компьютерным оборудованием.

2.2. Цели и задачи II модуля (базовый уровень)

Цель базового уровня: углубление навыков самостоятельного написания кода и разработки эффективных алгоритмов программирования на языке Python .

Для успешной реализации поставленной цели необходимо решить ряд педагогических, развивающих и воспитательных задач:

Обучающие:

- закрепить опыт объектно-ориентированного и функционального программирования в различных интегрированных средах разработки на языке Python;

- углубить и структурировать знания основ современных языков программирования;

- сформировать навыки разработки эффективных алгоритмов и программ на основе изученного языка программирования Python;

- сформировать навыки работы со специальными средствами и библиотеками языка Python, в том числе с графической библиотекой Tkinter;

- сформировать навыки написания грамотного, красивого кода, уметь находить и обрабатывать ошибки в коде;

- обучить практическому использованию сложных структур данных.

Развивающие:

- способствовать развитию творческих способностей обучающихся с использованием межпредметных связей (информатика, технология, окружающий мир, физика, математика);

- способствовать развитию исследовательской и проектной деятельности;

- способствовать формированию навыков планировать свои действия с учётом фактора времени, а также предвидеть результаты своей работы и оптимальные пути их достижения;

- способствовать развитию навыков самостоятельного изучения новых технологий.

Воспитательные:

- способствовать развитию основ коммуникативных отношений внутри проектных групп и в коллективе в целом;

- способствовать воспитанию упорства в достижении результата;

- способствовать формированию целеустремлённости, организованности, ответственного отношения к труду и уважительного отношения к окружающим;

– способствовать созданию условий для развития устойчивой потребности в самообразовании.

2.3. Цели и задачи III модуля (продвинутый уровень)

Цель базового уровня: формирование комплексных компетенций в области работы с данными и веб-разработки на языке Python .

Для успешной реализации поставленной цели необходимо решить ряд педагогических, развивающих и воспитательных задач:

Обучающие:

- познакомить с основными методами хранения информации: от CSV-таблиц до реляционных и нереляционных баз данных;
- освоить базовые операции SQL;
- развить навыки практического применения баз данных в реальных приложениях;
- научить использовать библиотеку requests для выполнения различных типов HTTP-запросов;
- освоить работу с форматом JSON и принципами сериализации данных;
- понять концепцию API и принципы взаимодействия между сервисами;
- научить создавать веб-серверы на базе Flask, использовать шаблонизаторы и работать с ORM;
- сформировать навыки проектной деятельности: от постановки задачи до защиты итогового проекта.

Развивающие:

- способствовать развитию системного, алгоритмического и логического мышления;
- способствовать развитию навыков анализа и структурирования информации, умения работать с технической документацией;
- способствовать развитию навыков исследовательской и проектной деятельности, креативности и инициативности при реализации проектов;

- способствовать формированию навыков планировать свои действия с учётом фактора времени, а также предвидеть результаты своей работы и оптимальные пути их достижения;

- способствовать развитию навыков самостоятельного поиска решений, навыков решения проблем при интеграции различных технологий, навыков изучения новых технологий.

Воспитательные:

- способствовать развитию коммуникативных отношений внутри проектных групп и в коллективе в целом;

- способствовать формированию ответственности за корректность и безопасность обработки данных;

- способствовать воспитанию аккуратности и внимания к деталям при написании кода, составлении запросов и настройке инфраструктуры;

- способствовать формированию этичного отношения к использованию данных и соблюдению авторских прав при работе с API и внешними сервисами;

- способствовать формированию целеустремлённости, организованности, ответственного отношения к труду и уважительного отношения к окружающим;

- способствовать созданию условий для развития устойчивой потребности в самообразовании.

3. Содержание общеразвивающей программы

3.1. Учебный план I модуля (стартовый уровень)

Таблица 1

№ п/п	Название раздела, темы	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
Раздел 1. Введение в программирование		34	9	25	
1.1	Вводное занятие. Инструктаж по ТБ. Знакомство со средой разработки. Антикоррупционное просвещение	2	1	1	Фронтальный опрос
1.2	Переменные, операторы, ввод и вывод данных. Типы данных.	8	2	6	Педагогическое наблюдение
1.3	Условные конструкции	10	2	8	Педагогическое наблюдение
1.4	Работа со строками	8	2	6	Педагогическое наблюдение
1.5	Решение задач по разделу «Введение в программирование».	6	2	4	Анализ работ
Раздел 2. Базовые конструкции в Python		36	8	28	
2.1	Цикл for	10	2	8	Педагогическое наблюдение
2.2	Цикл while	10	2	8	Педагогическое наблюдение
2.3	Работа со списками	8	2	6	Педагогическое наблюдение
2.4	Решение задач по разделу «Базовые конструкции в Python»	8	2	6	Анализ работ
Раздел 3. Функциональное программирование		54	16	38	
3.1	Работа со словарями	8	2	6	Педагогическое наблюдение
3.2	Множества	6	2	4	Педагогическое наблюдение
3.3	Двумерный массив	8	4	4	Педагогическое наблюдение
3.4	Функция и рекурсия	8	2	6	Педагогическое наблюдение

3.5	Работа с файлами	6	2	4	Педагогическое наблюдение
3.6	Основы объектно-ориентированного программирования	10	2	8	Педагогическое наблюдение
3.7	Решение задач по разделу «Функциональное программирование»	8	2	6	Анализ работ
Раздел 4. Проектная деятельность		20	4	16	
4.1	Применение гибкого управления проектами	2	1	1	Фронтальный опрос
4.2	Разработка MVP проекта	12	2	10	Анализ работ
4.3	Инструменты и методы эффективной презентации	4	1	3	Анализ работ
4.4	Итоговая защита проекта	2	0	2	Защита проекта
Итого		144	37	107	

Содержание учебного плана

Раздел 1. Введение в программирование

Тема 1.1 Вводное занятие. Инструктаж по ТБ. Знакомство со средой разработки. Антикоррупционное просвещение.

Теория: Инструктажи по технике безопасности. Беседа об антикоррупционном просвещении. Знакомство со средой разработки, видами ПО, отличительными особенностями языка.

Практика: Тренинг на знакомство. Установка среды разработки. Написание первой простейшей программы.

Тема 1.2 Переменные, операторы, ввод и вывод данных. Типы данных.

Теория: Переменные как ячейки данных, их использование и очистка данных. Ввод и вывод значений. Типы данных `int()` и `float()`. Операторы для вычислений. Библиотека `math`.

Практика: Решение задач.

Тема 1.3 Условные инструкции

Теория: Разбор таблицы истинности, сравнение двух значений и возвращение результата логического типа. Синтаксис условной инструкции.

Вложенные условные инструкции. Операторы сравнения. Тип данных bool.
Каскадные условные инструкции.

Практика: Решение задач.

Тема 1.4 Работа со строками.

Теория: Изучение типа данных «строки». Срезы. Методы.

Практика: Решение задач.

Тема 1.5 Решение задач по разделу «Введение в программирование».

Теория: Обобщение, систематизация теоретического материала для написания кода.

Практика: Решение задач по пройденным темам.

Раздел 2. Базовые конструкции в Python.

Тема 2.1 Цикл for.

Теория: Цикл for. Функция range. Настройка функции print().

Практика: Решение задач.

Тема 2.2 Цикл while.

Теория: Цикл while. Условия выполнения итераций и инструкции управления циклом. Проблемы заикливания и выход из него. Множественное присваивание.

Практика: Решение задач.

Тема 2.3 Работа со списками.

Теория: Массив данных, обращение к элементам по индексам. Срезы. Ввод и вывод данных массива. Генераторы списков. Методы split и join.

Практика: Решение задач.

Тема 2.4 Решение задач по разделу «Базовые конструкции в Python».

Теория: Подходы к решению задач и способы написания кода.

Практика: Обобщение, систематизация теоретического материала.

Подходы к решению задач и способы написания кода.

Раздел 3. Функциональное программирование.

Тема 3.1 Работа со словарями.

Теория: Словари: что такое ключ и значение. Когда нужно использовать словари. Создание словаря. Работа с элементами словаря и их перебор.

Практика: Решение задач.

Тема 3.2 Множества.

Теория: Что такое множества. Создание множеств. Работа с элементами множеств. Операции над множествами.

Практика: Решение задач.

Тема 3.3 Двумерный массив.

Теория: Что такое двумерный массив. Создание вложенных списков. Ввод двумерного массива. Пример обработки двумерного массива. Вывод вложенных списков и их элементов. Вложенные генераторы двумерных массивов.

Практика: Решение задач.

Тема 3.4 Функции и рекурсия.

Теория: Создание функций и обращение к ним. Локальные и глобальные переменные. Рекурсия.

Практика: Решение задач.

Тема 3.5 Работа с файлами.

Теория: Ввод и вывод данных в файл для последующего хранения. Чтение данных из файла.

Практика: Решение задач.

Тема 3.6 Основы объектно-ориентированного программирования.

Теория: Основные понятия и принципы ООП. Создание класса. Методы. Конструктор. Деструктор. Атрибуты класса (поля). Обращение к классу и атрибуту.

Практика: Решение задач.

Тема 3.7 Решение задач по разделу «Функциональное программирование»

Теория: Подходы к решению задач и способы написания кода.

Практика: Обобщение, систематизация теоретического материала.
Подходы к решению задач и способы написания кода.

Раздел 4. Проектная деятельность.

Тема 4.1 Применение гибкого управления проектами.

Теория: Методики управления проектами. Гибкая и каскадная модель управления. Характеристика проекта.

Практика: Мозговой штурм (генерация проектных идей). Деловая игра по проектному управлению. Практическая работа.

Тема 4.2 Разработка MVP проекта.

Теория: Что такое MVP и его типы. Как построить минимально жизнеспособный продукт. Примеры MVP в IT-отраслях.

Практика: Изучение успешных IT-проектов (кейс-метод). Практическая работа.

Тема 4.3 Инструменты и методы эффективной презентации.

Теория: Виды и продолжительность эффективной презентации. Структура презентации. Этапы ее создания. Сервисы и программы.

Практика: Практическая работа.

Тема 4.4 Итоговая защита проекта.

Практика: Защита проекта.

3.2. Учебный план II модуля (базовый уровень)

Таблица 2

№ п/п	Название раздела, темы	Кол-во часов			Формы аттестации/контроля
		Всего	Теория	Практика	
Раздел 1. Программирование на Python		60	18	42	
1.1	Вводное занятие. Введение в программу 2-го года обучения Инструктаж по ТБ. Беседа на тему «Что значит быть честным?»	2	2	0	Фронтальный опрос
1.2	Повторение. Решение задач на основные конструкции и структуры данных	8	2	6	Анализ работ
1.3	Объектно-ориентированное программирование	10	4	6	Педагогическое наблюдение
1.4	Работа с рекурсивными функциями. Промежуточный контроль	4	1	3	Педагогическое наблюдение
1.5	Работа с args, kwargs	8	2	6	Педагогическое наблюдение
1.6	Работа с модулем OS	8	2	6	Педагогическое наблюдение
1.7	Работа с исключениями	8	2	6	Педагогическое наблюдение
1.8	Использование пакетов и модулей	8	2	6	Педагогическое наблюдение
1.9	Решение задач по изученному материалу	4	1	3	Анализ работ
Раздел 2. Функциональное программирование		30	9	21	
2.1	Lambda-функции	6	2	4	Педагогическое наблюдение
2.2	Работа с функторами	6	2	4	Педагогическое наблюдение
2.3	Работа с декараторы	6	2	4	Педагогическое наблюдение

2.4	Регулярные выражения	8	2	6	Педагогическое наблюдение
2.5	Решение задач по изученному материалу.	4	1	3	Анализ работ
Раздел 3. Графическая библиотека Tkinter		32	8	24	
3.1	Графическая библиотека Tkinter: разработка простейшего приложения с графическим интерфейсом	6	2	4	Педагогическое наблюдение
3.2	Виджеты Listbox и Entry	6	1	5	Педагогическое наблюдение, анализ работ
3.3	Многооконные приложения	4	1	3	Педагогическое наблюдение, анализ работ
3.4	Виджет Canvas: Рисование двумерных фигур	4	1	3	Педагогическое наблюдение, анализ работ
3.5	Списки (массивы) виджетов	6	1	5	Педагогическое наблюдение, анализ работ
3.6	Пакет tkinter.ttk.	6	2	4	Педагогическое наблюдение
Раздел 4. Проектная деятельность.		22	4	18	
4.1	Применение гибкого управления проектами	2	1	1	Фронтальный опрос
4.2	Процесс разработки проекта	14	2	12	Анализ работ
4.3	Инструменты и методы эффективной презентации	4	1	3	Анализ работ
4.4	Итоговая защита проекта	2	0	2	Защита проекта
Итого		144	39	105	

Содержание учебного (тематического) плана

Раздел 1. Программирование на Python

Тема 1.1 Вводное занятие. Введение в программу 2-го года обучения

Инструктаж по ТБ. Беседа на тему «Что значит быть честным?».

Теория: Инструктаж по технике безопасности. Введение в программу.
Обсуждение профессий, использующих Python.

Тема 1.2 Повторение. Решение задач на основные конструкции и структуры данных.

Теория: Повторение основных конструкций и структур данных.

Практика: Входной контроль. Тренинг на командообразование.

Тема 1.3 Объектно-ориентированное программирование.

Теория: Актуализация знаний: Создание классов и объектов, методы, конструктор. Изучение нового: Области видимости. Перегрузка операторов. Основные принципы ООП. Отношения между классами (ассоциация: композиция и агрегация).

Практика: Решение задач.

Тема 1.4 Работа с рекурсивными функциями. Промежуточный контроль.

Теория: Создание рекурсивных функций, вызов функции и самовывоз внутри функции.

Практика: Решение задач.

Тема 1.5 Работа с args, kwargs.

Теория: Позиционные и именованные аргументы. Параметры по умолчанию. Оператор «звездочка». Комбинирование всех способов.

Практика: Решение задач.

Тема 1.6 Работа с модулем OS.

Теория: Модуль ОС. Функции модуля. Директория. Вычисление размера. Обработка путей.

Практика: Решение задач.

Тема 1.7 Работа с исключениями.

Теория: Механизм исключений. Встроенные исключения. Обработка исключений. Генерация исключений. Пользовательские исключения.

Практика: Решение задач.

Тема 1.8 Использование пакетов и модулей.

Теория: Модули и пакеты. Классификация модулей. Типы импортов: относительный и абсолютный импорт. Способы импорта. Виртуальное

окружение. Модуль `venv`. Переменная `__name__`

Практика: Решение задач.

Тема 1.9 Решение задач по изученному материалу.

Теория: Обобщение, систематизация теоретического материала.

Подходы к решению задач и способы написания кода.

Практика: Решение задач.

Раздел 2. Функциональное программирование

Тема 2.1 Lambda-функции.

Теория: Анонимная функция. Синтаксис lambda-функции. Lambda с условными выражениями. Интеграция с функциями высшего порядка (`map`, `filter`, `sorted`, `reduce`).

Практика: Решение задач.

Тема 2.2 Работа с функторами.

Теория: Функтор: что это такое и для чего. Сравнение с другими подходами. Функтора, как замена замыкания и частичного применения функции.

Практика: Решение задач.

Тема 2.3 Работа с декораторами.

Теория: Понимание функции как объекта. Функции, возвращающие функции. Определение декораторов. Декораторы с аргументами. Объекты-декораторы.

Практика: Решение задач.

Тема 2.4 Регулярные выражения.

Теория: Регулярные выражения: что это такое и для чего. Синтаксис `RegEx`. Область использования регулярных выражения. Модуль `re` и его методы.

Практика: Решение задач.

Тема 2.5 Решение задач по изученному материалу.

Теория: Обобщение, систематизация теоретического материала.

Подходы к решению задач и способы написания кода.

Практика: Решение задач.

Раздел 3. Графическая библиотека Tkinter

Тема 3.1 Графическая библиотека Tkinter: разработка простейшего приложения с графическим интерфейсом

Теория: Что такое GUI. Основные понятия. Создание окна. Общие свойства виджетов. Конфигурация главного окна приложения. Виджет Label. Менеджеры геометрии Pack. Настройка свойства font. Размещение изображения на метке. Виджет Button (кнопка). События и их обработка.

Практика: Решение задач.

Тема 3.2 Виджеты Listbox и Entry

Теория: Основные понятия: класс Listbox (список) и Entry (текстовое поле). Методы.

Практика: Решение кейса: Создание приложения «Ежедневник», который проверяет корректность введенных даты и времени, шаблон для ввода даты и времени, имеет кнопки: для очистки всех полей, сохранения данных в файл и загрузки данных из файла.

Тема 3.3 Многооконные приложения

Теория: Главные и вторичные окна. Алгоритм создания простого приложения с несколькими автономными окнами. Создание окон с использованием ООП.

Практика: Решение кейса: Создание приложения «Фотоальбом».

Тема 3.4 Виджет Canvas: Рисование двумерных фигур

Теория: Основные понятия. Параметры и методы виджета Canvas.

Практика: Решение задач.

Тема 3.5 Списки (массивы) виджетов

Теория: Основные понятия. Менеджер геометрии Grid, его параметры.

Практика: Решение кейса: Создание приложения «Калькулятор».

Тема 3.6 Пакет tkinter.ttk.

Теория: Пакет tkinter.ttk и его импорт. Виджет Notebook и его параметры. Виджет Radiobutton и его параметры. Окна сообщений: модуль

tkinter.messagebox и его функции.

Практика: Решение кейса: создание приложения «Тест».

Раздел 4. Проектная деятельность.

Тема 4.1 Применение гибкого управления проектами.

Теория: Методики управления проектами. Гибкая и каскадная модель управления. Характеристика проекта.

Практика: Мозговой штурм (генерация проектных идей). Практическая работа.

Тема 4.2 Процесс разработки проекта.

Теория: Проблематизация, построение гипотезы решения проблемы. Написание цели и задач по S.M.A.R.T. технологии.

Практика: Практическая работа.

Тема 4.3 Инструменты и методы эффективной презентации.

Теория: Виды и продолжительность эффективной презентации. Структура презентации. Этапы ее создания. Сервисы и программы.

Практика: Деловая игра (презентация проекта заказчику). Практическая работа.

Тема 4.4 Итоговая защита проекта.

Практика: Защита проекта.

3.3. Учебный план III модуля (продвинутый уровень)

Таблица 3

№ п/п	Название раздела, темы	Кол-во часов			Формы аттестации/контроля
		Всего	Теория	Практика	
Раздел 1. SQL		36	19	17	
1.1	Вводное занятие. Введение в программу 3-го года обучения Инструктаж по ТБ. Беседа на тему «Что значит быть честным?» Методы хранения информации	2	1	1	Фронтальный опрос
1.2	CSV таблицы, системы контроля версий	8	6	2	Анализ работ
1.3	SQL SELECT	4	2	2	Анализ работ
1.4	SQL INSERT	4	2	2	Анализ работ
1.5	SQL UPDATE	4	2	2	Анализ работ
1.6	SQL DELETE и нюансы Российского законодательства	4	2	2	Анализ работ
1.7	Минипроект. Применение БД в реальном приложении	10	4	6	Анализ работ
Раздел 2. WEB		60	28	32	
2.1	Как работает браузер, модель OSI	2	1	1	Педагогическое наблюдение
2.2	Библиотека requests. Типы запросов	2	1	1	Педагогическое наблюдение
2.3	Сериализация. JSON.	4	2	2	Педагогическое наблюдение
2.4	API. Как общаются машины?	2	1	1	Педагогическое наблюдение
2.5	Минипроект. Работа с использованием готового API	10	4	6	Педагогическое наблюдение, анализ работ

2.6	Веб-сервер. Flask	4	2	2	Педагогическое наблюдение, анализ работ
2.7	Шаблонизаторы	2	1	1	Педагогическое наблюдение, анализ работ
2.8	Знакомство с ORM	4	2	2	Педагогическое наблюдение, анализ работ
2.9	Blueprint в Flask	4	2	2	Педагогическое наблюдение, анализ работ
2.10	Собственное API, FastApi	8	4	4	Педагогическое наблюдение, анализ работ
2.11	Методы деплоя сервиса. Контейнеризация	4	2	2	Педагогическое наблюдение, анализ работ
2.12	Доступ извне. Тунели, сервера, домены	4	2	2	Педагогическое наблюдение, анализ работ
2.13	Минипроект. Веб-разработка	10	4	6	Анализ работ
Раздел 3. Проектная деятельность.		48	10	38	
3.1	Разработка проекта под потребности заказчика	8	4	4	Фронтальный опрос
3.2	Процесс разработки проекта	30	4	26	Анализ работ
3.3	Инструменты и методы эффективной презентации проекта	6	2	4	Анализ работ
3.4	Итоговая защита проекта	4	0	4	Защита проекта
Итого		144	57	87	

Содержание учебного (тематического) плана

Раздел 1. SQL

Тема 1.1 Вводное занятие. Введение в программу 3-го года обучения

Инструктаж по ТБ. Беседа на тему «Что значит быть честным?»

Методы хранения информации

Теория: Инструктаж по технике безопасности. Введение в программу. Обсуждение профессий, использующих Python. Беседа на тему «Что значит быть честным?» Формат табличных данных. Создание собственного формата. CSV таблицы. SQL базы данных.

Практика: Тренинг коммуникативных навыков, необходимых

айтишнику. Входной контроль.

Тема 1.2 CSV таблицы, системы контроля версий

Теория: Подход к коллективной разработке. Версионность кода, ветки и КОММИТЫ.

Практика: Решение задач.

Тема 1.3 SQL SELECT

Теория: Оператор для выборки данных из баз данных.

Практика: Решение задач.

Тема 1.4 SQL INSERT

Теория: Оператор для добавления строк в БД.

Практика: Решение задач.

Тема 1.5 SQL UPDATE

Теория: Оператор для изменения строк в БД.

Практика: Решение задач.

Тема 1.6 SQL DELETE и нюансы Российского законодательства

Теория: Оператор для удаления строк в БД. Соблюдение законодательства с области сохранения пользовательских данных. Теневое удаление.

Практика: Решение задач.

Тема 1.7 Минипроект. Применение БД в реальном приложении

Теория: Примеры успешных IT-проектов (кейс-метод).

Практика: Мозговой штурм (генерация идей). Разработка и презентация минипроекта.

Раздел 2. WEB

Тема 2.1 Как работает браузер, модель OSI

Теория: Как браузер загружает сайт. Модель OSI: От тока в проводах, до отправки GET запроса.

Практика: Решение задач.

Тема 2.2 Библиотека requests. Типы запросов

Теория: Типы запросов: GET, POST, PUT, DELETE. Программная

отправка таких запросов.

Практика: Решение задач.

Тема 2.3 Сериализация. JSON.

Теория: Что такое сериализация. Текстовое, структурированное представленные данных.

Практика: Решение задач.

Тема 2.4 API. Как общаются машины?

Теория: Что такое API. RestFull API. Единая точка бекенда.

Практика: Решение задач.

Тема 2.5 Минипроект. Работа с использованием готового API

Теория: Примеры успешных IT-проектов (кейс-метод).

Практика: Мозговой штурм (генерация идей). Разработка и презентация минипроекта.

Тема 2.6 Веб-сервер. Flask

Теория: Зачем нужен веб-сервер. Используемые порты. Возврат HTML пользователю.

Практика: Решение задач.

Тема 2.7 Шаблонизаторы

Теория: Отказ от дублирования кода. Общие содержимое HTML.

Практика: Решение задач.

Тема 2.8 Знакомство с ORM

Теория: Следующий шаг после SQL: БД как объект. Оверхед.

Практика: Решение задач.

Тема 2.9 Blueprint в Flask

Теория: Разбиение кода на отдельные модули.

Практика: Решение задач.

Тема 2.10 Собственное API, FastApi

Теория: Создание своего API. Принципы REST.

Практика: Решение задач.

Тема 2.11 Методы деплоя сервиса. Контейнеризация

Теория: Docker: создание контейнеров. Изоляция выполняющей среды.

Практика: Решение задач.

Тема 2.12 Доступ извне. Тунели, сервера, домены

Теория: Личный IP. Что такое NAT и способы его пробития. Сервисы тунелирования.

Практика: Решение задач.

Тема 2.13 Минипроект. Веб-разработка

Теория: Примеры успешных IT-проектов (кейс-метод).

Практика: Мозговой штурм (генерация идей). Разработка и презентация минипроекта.

Раздел 3. Проектная деятельность.

Тема 3.1 Разработка проекта под потребности заказчика.

Теория: Методы выявления потребностей заказчика. Формирование портрета конечного пользователя и анализ его задач (user stories, user personas). Документирование требований: составление брифа, технического задания (ТЗ). Подходы к монетизации.

Практика: Деловая игра по проектному управлению. Выбор методологии разработки: Agile (Scrum, Kanban), Waterfall, гибридные подходы.

Тема 3.2 Процесс разработки проекта.

Теория: Проблематизация, построение гипотезы решения проблемы. Написание цели и задач по S.M.A.R.T. технологии.

Практика: Практическая работа. Распределение ролей в команде (матрица RACI). Составление графика работ (диаграмма Ганта, дорожные карты).

Тема 3.3 Инструменты и методы эффективной презентации проекта.

Теория: Виды и продолжительность эффективной презентации. Структура презентации. Этапы ее создания. Сервисы и программы.

Практика: Изучение примеров эффективной презентации проекта

(кейс-метод). Практическая работа.

Тема 3.4 Итоговая защита проекта.

Практика: Защита проекта.

4. Планируемые результаты

Предметные результаты:

- знание основных терминов программирования и умение использовать их при решении практических задач;
- умение писать и отлаживать код на Python для решения различных задач;
- знание основ применения языка программирования в различных областях;
- умение написать эффективный и оптимизированный код;
- навык создания MVP проектов.

Метапредметные результаты:

- навыки планирования рабочей деятельности по реализации замысла, предвидения результата и его достижения, внесения корректировок в первоначальный замысел;
- навыки работы с различными источниками информации, умение самостоятельно искать, извлекать и отбирать необходимую информацию;
- знание правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой, санитарно-гигиеническими нормами;
- навыки изложения мысли в четкой логической последовательности, отстаивания точки зрения, анализа ситуации и самостоятельного поиска ответов, путем логических рассуждений;
- умение планирования создания продукта от идеи до действующего прототипа / макета, с учетом выстраивания междисциплинарных связей в области информатики, математики, физики, мехатроники.

Личностные результаты:

- способность доброжелательно относиться к окружающим в процессе групповой работы, выстраивать отношения делового сотрудничества, взаимоуважения;

- понимание необходимости уважительного отношения к другому человеку, его мнению и деятельности;
- умение ответственно относиться к учению и труду, проявление упорства в достижении результата;
- умение работать в группе и коллективе в процессе проектной и учебно-исследовательской деятельности.

Планируемые результаты I модуля (стартовый уровень)

Предметные результаты:

- умение работать с основными конструкциями языка программирования;
- уметь пользоваться комплексом базовых понятий и принципов функционального и объектно-ориентированного программирования (знание структур данных, базовые принципы их обработки);
- умение решать прикладные задачи на языке Python;
- умение программировать оборудование с помощью Python;
- умение работать с информацией необходимой для программирования на языке Python (поиск, анализ, использование информации в сети интернет);
- навык работы с компьютером и прикладными программами.

Метапредметные результаты:

- умение самостоятельно планировать последовательность своих действий для достижения поставленных целей, а также грамотно распределять свое время и ресурсы для получения максимально эффективного результата;
- умение осуществлять самостоятельный поиск информации, анализировать и обобщать её;
- проявление интереса к сфере программирования и техническим видам творчества;
- способность к принятию решений, а также умение формулировать, аргументировать и отстаивать своё мнение
- знание правил поведения при работе с компьютерной техникой.

Личностные результаты:

- соблюдение правил техники безопасности при работе с компьютерной техникой;
- проявление усидчивости и внимательности во время образовательного процесса;
- демонстрирует позитивное отношение к другому человеку, его мнению, результату его деятельности.

Планируемые результаты II модуля (базовый уровень)

Предметные результаты:

- умение разрабатывать и использовать компьютерно-математические модели и использовать на практике сложные структуры данных;
- умение работать со специальными средствами и библиотеками языка Python;
- знание основ современных языков программирования;
- знание базовых управляющих конструкций объектно-ориентированного программирования и умение их использовать;
- умение самостоятельно грамотно создавать программы анализа данных, находить ошибки в коде;
- навык разработки эффективных алгоритмов.

Метапредметные результаты:

- умение использовать междисциплинарные связи;
- умение проводить исследование в интересующей области;
- умение планировать последовательность шагов для достижения целей, соотношение своих действий с планируемыми результатами, осуществление контроля своей деятельности в процессе достижения результата;
- самостоятельность в изучении новых технологий.

Личностные результаты:

- активно вступает в диалог, ведет диалог с учетом общепринятых норм эффективной коммуникации;
- демонстрирует позитивное отношение к другому человеку, его мнению, результату его деятельности;
- проявляет устойчивый интерес к саморазвитию;
- проявляет упорство в достижении результата.

Планируемые результаты III модуля (продвинутый уровень)

Предметные результаты:

- знание основных методов хранения информации;
- знание базовых операций SQL;
- навыки практического применения баз данных в реальных приложениях;
- умение использовать библиотеку requests для выполнения различных типов HTTP-запросов;
- умение работать с форматом JSON и принципами сериализации данных;
- знание концепции API и принципов взаимодействия между сервисами;
- умение создавать веб-серверы на базе Flask, использовать шаблонизаторы и работать с ORM;
- навыки проектной деятельности: от постановки задачи до защиты итогового проекта.

Метапредметные результаты:

- проявление системного, алгоритмического и логического мышления;
- демонстрацию навыков анализа и структурирования информации, умения работать с технической документацией;
- применение навыков исследовательской и проектной деятельности, креативности и инициативности при реализации проектов;

- применение навыков планировать свои действия с учётом фактора времени, а также предвидеть результаты своей работы и оптимальные пути их достижения;

- применение навыков самостоятельного поиска решений, навыков решения проблем при интеграции различных технологий, навыков изучения новых технологий.

Личностные результаты:

- демонстрирует коммуникативные навыки внутри проектных групп и в коллективе в целом;

- проявляет ответственность за корректность и безопасность обработки данных;

- демонстрирует аккуратность и внимание к деталям при написании кода, составлении запросов и настройке инфраструктуры;

- демонстрирует этичное отношение к использованию данных и соблюдению авторских прав при работе с API и внешними сервисами;

- демонстрирует целеустремлённость, организованность, ответственное отношение к труду и уважительное отношение к окружающим;

- проявляет устойчивую потребность в самообразовании.

5. ВОСПИТАНИЕ

5.1. Цель, задачи, целевые ориентиры воспитания

Целью воспитания является обеспечение личностного развития обучающихся, формирование гражданских, патриотических и нравственных качеств личности.

Задачи:

1. Личностное развитие: формирование самосознания, уверенности в себе, развитие эмоционального интеллекта и социальных навыков.
2. Креативность и критическое мышление: стимулирование творческого подхода к решению задач, развитие аналитических навыков и способности к критическому осмыслению информации.
3. Социальная адаптация: помощь в интеграции в общество, развитие навыков общения, умение работать в команде и взаимодействовать с разными людьми.
4. Ценностное ориентирование: формирование системы ценностей, моральных и этических норм, понимание ответственности за свои действия.
5. Профессиональная ориентация: ознакомление с различными профессиями и направлениями, что поможет обучающимся определиться с будущей карьерой.

Воспитание в дополнительном образовании должно быть комплексным и индивидуализированным, учитывая уникальные потребности и интересы каждого подростка.

Основные целевые ориентиры воспитания направлены на формирование:

- интереса к технической деятельности, истории техники в России и мире, к достижениям российской и мировой технической мысли;
- понимание значения техники и информационных технологий в жизни российского общества;
- ценности авторства и участия в техническом творчестве;

- навыка определения достоверности и этики технических идей и информационных продуктов;
- ценности технической и информационной безопасности и контроля;
- уважения к достижениям своих земляков в технической сфере и сфере информационных технологий;
- воли, упорства, дисциплинированности в реализации проектов;
- опыта участия в технических проектах.

Эти ориентиры помогут создать всестороннее развитие у детей в области ИТ, подготовив их к вызовам будущего и формируя навыки, необходимые для успешной жизни и карьеры.

5.2. Формы и методы воспитания

Основной формой воспитания и обучения детей в системе дополнительного образования технической направленности является учебное занятие. В ходе учебных занятий в соответствии с предметным и метапредметным содержанием программ обучающиеся: усваивают информацию, имеющую воспитательное значение; получают опыт деятельности, в которой формируются и проявляются ценностные, нравственные ориентации; участвуют в освоении и формировании среды своего личностного развития, творческой самореализации.

Формы и методы воспитания обучающихся 12-17 лет должны учитывать особенности подросткового и юношеского возраста, характеризующегося интенсивным физическим и психическим развитием, формированием самосознания и личностной идентичности, поиском своего места в обществе и усилением влияния сверстников.

На занятиях применяются следующие *формы воспитания*:

Проектная деятельность: работа над проектами способствует развитию самостоятельности, ответственности, критического мышления и умения работать в команде. Участие в проектах и соревнованиях способствует формированию умений в области целеполагания, планирования

и рефлексии, укрепляет внутреннюю дисциплину, даёт опыт долгосрочной системной деятельности.

Дебаты и дискуссии направлены на развитие критического мышления, умения аргументировать свою позицию, толерантности и уважения к мнению других.

Экскурсии и поездки на предприятия способствуют расширению кругозора, познавательному развитию, формированию гражданской позиции.

Соревнования, конкурсы: стимулирование творческого самовыражения, развитие самоуважения и уверенности в себе.

Тренинги и мастер-классы: развитие конкретных навыков (коммуникативных, личностных, профессиональных), помощь в самопознании и саморазвитии.

Методы воспитания:

Проблемное обучение: стимулирует самостоятельный поиск решения задач, развивает критическое мышление и аналитические способности.

Дискуссионные методы: позволяют обсуждать актуальные проблемы, формировать собственное мнение и отстаивать его.

Исследовательские методы: развивают навыки исследовательской работы, позволяют глубоко изучить интересующую тему.

Методы личностно-ориентированного обучения: учёт индивидуальных особенностей и потребностей обучающихся.

Информационно-коммуникативные технологии: использование современных технологий для повышения эффективности образовательного процесса.

Методы положительного подкрепления: поощрение успехов и достижений стимулирует дальнейшее развитие и самосовершенствование.

Важно учитывать потребность подростков в автономии и самостоятельности, стремление к самоутверждению и принятию со стороны сверстников. Методы воспитания должны быть направлены на формирование

ответственности, самостоятельности, критического мышления, толерантности и умения эффективно взаимодействовать с окружающими.

5.3. Условия воспитания, анализ результатов

Воспитательный процесс осуществляется в условиях организации деятельности обучающихся на основной учебной базе реализации программы в организации дополнительного образования детей в соответствии с нормами и правилами работы этой организации.

Для всестороннего анализа эффективности воспитательной работы используются следующие комплексные показатели, охватывающие различные аспекты развития личности и социальной адаптации обучающихся:

1. Личностный рост: этот показатель оценивает глубинные изменения в личности обучающихся. Анализ проводится по следующим направлениям:

Развитие ответственности: оценивается способность принимать решения, прогнозировать последствия своих действий, самостоятельно выполнять свои обязанности, нести ответственность за порученное дело и его результаты.

Развитие самостоятельности: оценивается способность к самоорганизации, принятию решений без постоянного контроля со стороны взрослых, умение планировать свою деятельность и достигать поставленных целей.

Развитие коммуникативных навыков: оценивается способность эффективно взаимодействовать с окружающими, строить конструктивные отношения, выражать свои мысли и чувства, слушать и понимать других.

Методами оценки служат наблюдение за поведением в различных ситуациях (учебной, внеучебной), анализ выполнения заданий, самооценка, анкетирование, ролевые игры, анализ участия в дискуссиях, анализ метапредметных и личностных компетенций.

2. Достижения: этот показатель отражает успехи обучающихся в различных областях деятельности (участие в конкурсах, хакатонах,

соревнованиях). Оцениваются не только победы, но и сам процесс достижения результата, проявленные усилия, настойчивость.

3. Социальная активность: этот показатель характеризует степень вовлеченности обучающихся в социальную жизнь коллектива и общества (участие в коллективных делах; волонтерская деятельность).

4. Удовлетворенность обучающихся: проводятся анкетирования и интервью, с целью выявления степени удовлетворенности обучающихся и их родителей воспитательной работой, предложения по её улучшению.

Комплексный анализ всех перечисленных показателей позволяет получить объективную картину эффективности воспитательной работы и внести необходимые коррективы в воспитательный процесс.

Методами анализа могут служить:

1. Педагогическое наблюдение, в процессе которого внимание педагогов сосредотачивается на проявлении в деятельности детей и в её результатах определённых в данной программе целевых ориентиров воспитания, а также на проблемах и трудностях достижения воспитательных задач программы;

2. Оценка творческих и исследовательских работ и проектов экспертным сообществом (педагоги, родители, приглашённые внешние эксперты и др.) с точки зрения достижения воспитательных результатов;

3. Самоанализ и самооценка обучающихся по итогам деятельности, отзывы родителей (законных представителей) и других участников образовательных событий и мероприятий.

Эффективность воспитания зависит от системного подхода, учета индивидуальных потребностей, взаимодействия с семьей, создания условий для самореализации. Результаты воспитания проявляются в личностном росте, социальной адаптации и достижениях обучающихся. Анализ результатов позволяет корректировать программы и повышать их эффективность.

5.4. Календарный план воспитательной работы

№ п/п	Название события, мероприятия	Сроки	Форма проведения	Ответственный
1	Мероприятие антикоррупционной направленности	сентябрь 2026	очно (беседа, викторина)	Педагог- организатор
2	Инструктажи по технике безопасности, пожарной, антитеррористической безопасности	сентябрь 2026	очно	Педагог дополнительного образования
3	Игры на знакомство и командообразование в учебных группах	октябрь 2026	очно (тренинг)	Педагог- организатор
4	Мастер-класс «Что такое проект: как найти тему и что может стать результатом»	декабрь 2026	мастер-класс	Педагог дополнительного образования
5	«Дни науки»	февраль 2027	мастер-классы, игры	Педагог- организатор
6	Комплекс мероприятий к Дню защитника Отечества	февраль 2027	игры, мастер- классы, викторины	Педагог- организатор
7	Комплекс мероприятий к Дню космонавтики	март 2027	игры, мастер- классы, викторины	Педагог- организатор
8	Комплекс мероприятий ко Дню Победы в IT- куб	май 2027	игры, мастер- классы, викторины	Педагог- организатор
9	Итоговая защита проектов обучающихся	май 2027	очная защита проектов	Педагог дополнительного образования
10	Информирование и привлечение обучающихся к участию в конкурсных мероприятиях разного уровня	в течение года	конкурсы, соревнования, хакатоны, олимпиады	Педагог- организатор
11	Информирование и привлечение обучающихся к участию в экскурсиях на промышленные предприятия района и города	в течение года	экскурсии	Педагог- организатор
12	Информирование и привлечение обучающихся к участию в мероприятиях IT- куб	в течение года	конкурсы, соревнования, хакатоны, олимпиады	Педагог- организатор

**II. Комплекс организационно-педагогических
условий реализации общеразвивающей
программы**

1.Календарный учебный график на 2026–2027 учебный год

Таблица 5

№ п/п	Основные характеристики образовательного процесса	
1.	Количество учебных недель	36
2.	Количество учебных дней	36
3.	Количество часов в неделю	4
4.	Количество часов в год	144
5.	Недель в I полугодии	16
6.	Недель в II полугодии	20
7.	Начало занятий	07 сентября
8.	Выходные дни	31 декабря – 10 января
9.	Окончание учебного года	31 мая

2. Условия реализации программы

2.1 Материально-техническое обеспечение

Учебные аудитории, отвечающие требованиям СП 2.4.3648-20 для учреждений дополнительного образования, с индивидуальными рабочими местами (столы, стулья) обучающихся и одним рабочим местом для педагога дополнительного образования.

Требования к помещению:

- помещение для занятий, отвечающие требованиям СП 2.4.3648–20 для учреждений дополнительного образования;
- качественное освещение;
- столы, стулья по количеству обучающихся и 1 рабочим местом для педагога;
- напольная мобильная стойка для интерактивных досок или универсальное настенное крепление;
- доска магнито-маркерная настенная;
- флипчарт.

Расходные материалы:

- whiteboard маркеры;
- стиратель с доски;
- бумага писчая;
- простые карандаши;
- шариковые ручки;
- permanent маркеры.

Оборудование:

- ноутбуки для каждого обучающегося и преподавателя;
- многофункциональное устройство (принтер, сканер и копир);
- моноблочное интерактивное устройство;
- Интерактивная панель (комплекс) NextPanel 75S KR 203 AA-W1
- Стационарный компьютер;
- Сетевое устройство;
- Web-камера;

– Наушники.

Информационное обеспечение:

– Python;

– операционная система (Astra Linux);

– поддерживаемые браузеры (для работы LMS): Yandex Browser;

– среда Wing IDE 101 (версии 6 или выше);

– среда PyCharm, среда Visual Studio;

– пакет PyQt4 (на Qt5);

– пакет библиотек со SciPy: numpy, scipy, matplotlib, ipython +
ipythonnotebook, sympy, pandas.

2.2. Кадровое обеспечение

Теоретические и практические занятия реализуются педагогом дополнительного образования, обладающим профессиональными знаниями и компетенциями в организации и проведении образовательной деятельности согласно содержанию ДООП.

Уровень образования: среднее профессиональное образование, высшее образование – бакалавриат, специалитет или магистратура. Уровень соответствия квалификации: образование педагога соответствует профилю ДООП. Профессиональная категория: без требований к категории.

2.3 Формы аттестации и оценочные материалы

Система контроля знаний и умений обучающихся представляется в виде учёта индивидуального результата по итогам выполнения практических заданий и посредством наблюдения, отслеживания динамики развития обучающегося, по каждому контрольному мероприятию и подведения в итоге суммарного балла для каждого обучающегося.

Система отслеживания результатов обучающихся выстроена следующим образом:

- входная диагностика;
- промежуточный контроль;
- итоговый контроль.

Входная диагностика по программе проводится в виде тестирования. (Приложение 4).

Оценивая личностные и метапредметные результаты обучающихся, педагог проводит наблюдение за обучающимися, отслеживание динамики изменения их творческих, коммуникативных и иных способностей (Приложение 1, 2).

Текущий контроль осуществляется регулярно во время занятий. Контроль теоретических знаний осуществляется с помощью педагогического наблюдения, опросов, решения задач, практических работ. В практической деятельности результативность оценивается качеством выполнения работ обучающихся, где анализируются положительные и отрицательные стороны работ, корректируются недостатки.

Система промежуточного и итогового контроля знаний и умений обучающихся представляется в виде учёта индивидуального результата по каждому контрольному мероприятию и подведения в итоге суммарного балла для каждого обучающегося.

Промежуточный контроль проводится в форме тестирования обучающихся (Приложение 5). Максимальное количество баллов, которое возможно получить по результатам промежуточного контроля – 40 баллов.

Итоговый контроль обучающихся реализуется посредством оценки итоговых проектов. Для этого педагог заполняет предложенный лист (Приложение 3). Максимальное количество баллов за выполнение итогового проекта – 60 баллов.

Степень освоения программы оценивается в конце обучения (сумма баллов результатов промежуточного контроля и защиты итогового проекта). Оценка осуществляется по стобальной шкале, которая переводится в один из уровней освоения образовательной программы согласно таблице 6:

Уровень освоения программы по окончании обучения

Таблица 6

Баллы	Уровень освоения программы
0–39 баллов	Низкий
40–79 баллов	Средний
80-100 баллов	Высокий

2.4. Методические материалы

Образовательный процесс осуществляется в очной форме с применением электронного обучения и дистанционных образовательных технологий.

Виды занятий с использованием дистанционных технологий: видеоконференция, вебинар, видеоурок. Платформы трансляции материала и организации взаимодействия: Мах, Сферум, Telemost, ВКонтакте.

В образовательном процессе используются следующие **методы**:

- объяснительно-иллюстративный;
- метод проблемного изложения (постановка проблемы и решение её самостоятельно или группой);
- проектно-исследовательский;
- наглядный (демонстрация плакатов, схем, таблиц, диаграмм; использование технических средств; просмотр видеоматериалов);
- практический (практические задания; анализ и решение проблемных ситуаций и т. д.).

Выбор методов обучения осуществляется исходя из анализа уровня готовности обучающихся к освоению содержания модуля, степени сложности материала, типа учебного занятия. На выбор методов обучения значительно влияет персональный состав группы, индивидуальные особенности, возможности и запросы детей.

Учитывая подростковый возраст обучающихся, потребность в постоянной мотивации обучающихся на занятиях, отдельные занятия имеют нетрадиционную структуру и проводятся с использованием современных методов обучения. К таким методам можно отнести применение тренинговых технологий, кейс-метод, метод деловой игры, метод взаимообучения, проектный метод, мозговой штурм.

В начале изучения курса на занятиях применяются тренинговые технологии (игры на знакомство, командообразование, эффективную коммуникацию). Тренинг считается одним из лучших способов налаживания

психологического климата и прохождения периода адаптации.

Цель тренинговых занятий:

- Совершенствование навыков общения, повышение уверенности в себе;
- Снятие барьеров, мешающих реальным и продуктивным действиям;
- Овладение приемами межличностного взаимодействия для повышения его эффективности.

Метод взаимообучения. Обучающиеся совместно в режиме микрогрупп ищут решение учебных проблем, задач, и в конце каждой работы участники проставляют самим себе оценки за активность и участие.

Проектный метод. При проектном методе обучающиеся сами ставят цель и определяют пути достижения. Они должны самостоятельно искать, отбирать, обобщать и анализировать данные. Задача преподавателя помогать обучающимся, отвечать на вопросы, консультировать.

Проектный метод решает следующие задачи:

- Учит обучающихся самостоятельно ставить цель и достигать ее.
- Формирование навыков работы с данными.
- Обучение применению полученных знаний на практике.
- Опыт в проведении исследований и презентации результата.

Кейс-метод. Метод изучения и анализа реальных ситуаций, которые могут иметь место в изучаемом направлении. В процессе обучающиеся учатся применять полученные теоретические знания на практике, правильно оценивать факты и соотносить их со своими знаниями, заявлять и аргументировать свою позицию. Для решения кейса нужно произвести анализ предложенной ситуации и оптимальное решение.

Мозговой штурм. Метод предполагает коллективное обсуждение поставленной задачи. Здесь обучающиеся свободно высказывают свои мнения и идеи. Задача педагога поставить цель, объяснить цели и правила.

Обучающимся нужно записывать возникающие идеи, которые затем подлежат совместному обсуждению. Данный метод помогает развивать интерес обучающихся, вырабатывать умения решения нестандартных задач, развивать навыки общения в коллективе.

Деловая игра. Метод состоит в моделировании всевозможных ситуаций и сторон изучаемого направления деятельности. Она используется для перевода полученных теоретических знаний в практическую область. Данный метод позволяет развивать у обучающихся нужные компетенции, анализировать процессы и взаимодействия, развивать коммуникативные навыки.

Методы воспитания: мотивация, убеждение, поощрение, упражнение, стимулирование, создание ситуации успеха и др.

Педагогические технологии: индивидуализации обучения; группового обучения; коллективного взаимообучения; дифференцированного обучения; разноуровневого обучения; проблемного обучения; развивающего обучения; дистанционного обучения; игровой деятельности; коммуникативная технология обучения; коллективной творческой деятельности; решения изобретательских задач; здоровьесберегающая технология.

Здоровьесберегающая деятельность реализуется:

- через создание безопасных материально-технических условий;
- через включение в занятие динамических пауз, периодической смены деятельности обучающихся;
- через контроль педагога за соблюдением обучающимися правил работы за ПК;
- через создание благоприятного психологического климата в учебной группе в целом.

Формы подведения итогов по отдельным темам программы и по итогам реализации общеразвивающей программы: тестирование, мониторинг, практическая работа, защита индивидуальных проектов.

Дидактические материалы:

– Методические пособия, разработанные преподавателем с учётом конкретных задач, варианты демонстрационных программ, материалы по терминологии ПО, учебная литература.

Список литературы

Литература, использованная при составлении программы:

1. Банкрашков, А.В. Программирование для детей на языке Python. Издательство: АСТ, 2017. – 96 с.
2. Изучаем Python: программирование игр, визуализация данных, веб приложения. 3-е изд. — СПб.: Питер, 2022.
3. Как устроен Python. Гид для разработчиков, программистов и интересующихся. — СПб.: Питер, 2021.
4. Мэтиз, Э. Изучаем Python: программирование игр, визуализация данных, веб-приложения. 3-е изд. — СПб.: Питер, 2020. – 216 с.
5. Лутц, М. Изучаем Python, том 1, 5-е изд.: Пер. с англ. — СПб.: ООО «Диалектика», 2019. – 235 с.
6. Свейгарт, Э. Автоматизация рутинных задач с помощью Python, 2-е изд.: Пер. с англ.—СПб.: ООО «Диалектика», 2021. – 140 с.
7. Томашевский, П. Привет, Python! Моя первая книга по программированию. Издательство: Наука и Техника, 2018. – 256 с.

Литература для родителей и обучающихся:

1. Воган Л., «Непрактичный» Python занимательные проекты для тех, кто хочет поумнеть. – СПб.: БХВ-Петербург, 2021. – 457 с.
2. Дэвид Копец. Классические задачи Computer Science на языке Python –СПб.: Питер, 2022 – 224 с.
3. Спрингер В., Гид по Computer Science для каждого программиста. – СПб.: Питер, 2020 – 193 с.
4. Таненбаум Эндрю, Бос Херберт. Современные операционные системы. – СПб.: Питер, 2022 – 1120 с.
5. Джейми Чан. Python Быстрый старт, 352 стр. 2021 г. – СПб.: Питер, 2022 – 224 с.
6. Питонтьютор. Бесплатный курс по программированию с нуля. // [Электронный ресурс] URL: <https://pythontutor.ru/> (дата обращения: 14.03.2026).

7. Code Basics: обучение базовым аспектам языков программирования от образовательной платформы Hexlet. // [Электронный ресурс] URL: <https://ru.codebasics.com/> (дата обращения: 20.03.2026).

Приложение 1

Мониторинг достижения обучающимися личностных результатов за 20 -20 год

№ п/п	Ф.И. обучающегося	Способность доброжелательно относиться к окружающим в процессе групповой работы, выстраивать отношения делового сотрудничества, взаимоуважения			Понимание необходимости уважительного отношения к другому человеку, его мнению и деятельности			Умение ответственно относиться к учению и труду, проявление упорства в достижении результата			Умение работать в группе и коллективе в процессе проектной и учебно-исследовательской деятельности		
		Входной	Промежуточный	Итоговый	Входной	Промежуточный	Итоговый	Входной	Промежуточный	Итоговый	Входной	Промежуточный	Итоговый
1													
2													
3													
4													
5													

3 балла – качество проявляется систематически
 2 балла – качество проявляется ситуативно
 1 балл – качество не проявляется

Значение показателя по группе:

1-1.7 – низкий уровень развития качества в группе
 1.8-2.5 – средний уровень развития качества в группе
 2.6-3 – высокий уровень развития качества в группе

**Мониторинг достижения обучающимися метапредметных результатов
за 20__-20__ год**

№ п/п	Ф.И. обучающегося	Навыки планирования деятельности по реализации замысла, предвидения результата и его достижения, внесения корректировок			Навыки работы с различными источниками информации, умение самостоятельно искать, извлекать и отбирать необходимую информацию			Знание правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой, санитарно-гигиеническими нормами			Навыки изложения мысли в четкой логической последовательности, отстаивания точки зрения, анализа ситуации и самостоятельного поиска ответов, путем логических рассуждений			Умение планирования создания продукта от идеи до действующего прототипа / макета, с учетом выстраивания межпредметных связей		
		Входной	Промежуточный	Итоговый	Входной	Промежуточный	Итоговый	Входной	Промежуточный	Итоговый	Входной	Промежуточный	Итоговый	Входной	Промежуточный	Итоговый
1																
2																
3																
4																
5																

Значение показателя по группе:

3 балла – качество проявляется систематически
2 балла – качество проявляется ситуативно
1 балл – качество не проявляется

1-1.7 – низкий уровень развития качества в группе
1.8-2.5 – средний уровень развития качества в группе
2.6-3 – высокий уровень развития качества в группе

Лист оценки итогового проекта (максимум 60 баллов)

№ группы: _____ Дата: _____
ФИО эксперта _____

№ п/п	ФИО обучающегося	Название проекта	Актуальность проекта, постановка проблемы, целеполагание (0–10 б.)	Соответствие написанной программы заданным целям (0–10 б.)	Степень владения специальными терминами (0–10 б.)	Практическая реализация, качество результата (0–10 б.)	Степень увлечённости процессом и стремления к оригинальности при выполнении заданий (0–10 б.)	Защита проекта (представление работы): презентация продукта, обоснование своей точки зрения, ответы на вопросы (0–10 б.)	Итого
1.									
2.									
3.									
4.									
5.									

Пример входной диагностики

(стартовый уровень)

(Максимальное количество - 10 баллов)

1) Язык программирования Python подходит для разработки: (2 балла)

- 1) Ничего из этого
- 2) Компьютерных и мобильных приложений
- 3) Все вышеперечисленное
- 4) Аналитика и машинное обучение
- 5) Игр

2) Что хранит в себе переменная? (2 балла)

- 1) Тип
- 2) Имя
- 3) Длину своего значения
- 4) Значение

3) $n = 4$ орудия независимо выстрелили по цели. Вероятность попадания в цель для каждого орудия равна $p = 0.4$. Найти вероятность, что в цель попадёт точно одно орудие (3 балла).

4) Какое из следующих чисел делится на 9? (3 балла)

- 1) 928
- 2) 981
- 3) 289
- 4) 988

Пример входной диагностики

(базовый уровень)

(Максимальное количество - 20 баллов)

1. Язык программирования Python подходит для разработки: (1 балл)

- а) Ничего из этого

- b) Компьютерных и мобильных приложений
- c) Все вышеперечисленное
- d) Аналитика и машинное обучение
- e) Игр

2. Что хранит в себе переменная? (1 балл)

- a) Тип
- b) Имя
- c) Длину своего значения
- d) Значение

3. Что обозначает тип данных int? (2 балла)

- a) Вещественное
- b) Строковое
- c) Логическое
- d) Целочисленное

4. Выберите правильную запись оператора присваивания (1 балл)

- a) $10 = x$
- b) $y = 7,8$
- c) $x = 5$
- d) $a == b + x$

5. Укажите оператор ввода: (1 балл)

- a) input()
- b) random()
- c) int()
- d) print()

6. Сколько возможных значений у переменной типа bool? (2 балла)

- a) Бесконечность не предел
- b) 10
- c) 2
- d) 4

7. Оператор цикла в языке Python: (3 балла)

- a) print
- b) while
- c) if
- d) for

8. Для чего нужен оператор break? (2 балла)

- a) Для поломки компьютера
- b) Для удаления программы
- c) Для выхода из цикла
- d) Для завершения программы

9. Напишите программу которая принимает 1 число и выводит 3 следующих за ним числа (3 балла)

10. Напишите программу которая принимает 10 различных чисел и выводит 2 списка: отрицательных и положительных (4 балла)

Пример входной диагностики

(продвинутый уровень)

(Максимальное количество - 10 баллов)

1. Тема: Объектно-ориентированное программирование (1 балл)

Какой метод является конструктором класса в Python, который автоматически вызывается при создании объекта?

- A) __init__
- B) __start__
- C) __constructor__
- D) __new_object__

2. Тема: Обработка исключений (1 балл)

Какой блок кода используется для перехвата и обработки ошибок, возникших в блоке try?

- A) catch
- B) except
- C) error

D) finally

3. Тема: Аргументы функций (1 балл)

Что позволяет сделать оператор `*args` в определении функции?

- A) Передать именованные аргументы в виде словаря
- B) Передать неограниченное количество позиционных аргументов (кортеж)
- C) Установить аргумент по умолчанию
- D) Запретить передачу аргументов

4. Тема: Модули и пакеты (1 балл)

Для чего используется переменная `__name__` в модуле Python?

- A) Для хранения имени автора кода
- B) Для проверки, запущен ли модуль как главный файл или импортирован
- C) Для автоматического импорта всех библиотек
- D) Для указания версии интерпретатора

5. Тема: Функциональное программирование (1 балл)

Как правильно записать `lambda`-функцию, которая возводит число `x` в квадрат?

- A) `lambda x: x * 2`
- B) `def lambda x: x ** 2`
- C) `lambda x: x ** 2`
- D) `function(x) = x * x`

6. Тема: Декораторы (1 балл)

Какой символ используется для применения декоратора к функции в Python?

- A) `#`
- B) `@`
- C) `&`
- D) `$`

7. Тема: Регулярные выражения (1 балл)

Какой стандартный модуль Python используется для работы с регулярными выражениями?

- A) `regex`

- B) pattern
- C) re
- D) match

8. Тема: Графический интерфейс Tkinter (1 балл)

Какой виджет в Tkinter предназначен для ввода однострочного текста пользователем?

- A) Label
- B) Button
- C) Entry
- D) Listbox

9. Тема: Менеджеры геометрии Tkinter (1 балл)

Какой менеджер геометрии позволяет размещать виджеты в виде таблицы (строки и столбцы)?

- A) pack
- B) place
- C) grid
- D) table

10. Тема: Работа с ОС и файлами (1 балл)

Какая функция модуля os.path рекомендуется для безопасного объединения путей к файлам с учетом операционной системы?

- A) os.path.combine
- B) os.path.join
- C) os.path.merge
- D) os.path.add

Пример промежуточной аттестации (стартовый уровень)

(Максимальное количество - 40 баллов)

- 1) Вывести в одну строку "меня зовут name", и "я люблю играть в game", где переменные name и game это ваше имя и игра, в которую вы любите играть (5 баллов).
- 2) Рассчитать сложность врага. Сложность врага рассчитывается из того за сколько ходов его можно победить. 1-5 легкий враг, 5-10 сложный враг, 10+ сложный враг. У врага есть hp, которое определяется случайно в диапазоне от 20 до 120. У игрока есть урон, который равен - 10. Программа должна выводить hp врага и его сложность в одну строчку (5 баллов).
- 3) Непросто приходится родителям капризной девочки Жени. Прошлым летом в июле она побывала в Туле, а в августе — в Пензе, и ей очень понравилось. Поэтому этим летом она снова хочет съездить в два различных города. При этом Женя хочет снова побывать в июле в Туле или в августе в Пензе, но не то и другое одновременно — повторять прошлогодний маршрут полностью ей будет скучно. Определите, подходит ли предлагаемый маршрут под требования Жени. Вводятся две строки — названия городов, в которые родители собираются отправиться с Женей в июле и в августе (5 баллов).
- 4) Удава можно измерять в попугаях. Для этого даже не нужно их глотать. Измерьте длину строки в попугаях. Напишите программу, которая определяет, сколько раз слово parrot укладывается в введенной строке (5 баллов).
- 5) Добрые друзья убеждают Буратино, что стоит ему посадить свой золотой на Поле чудес, как вырастет дерево с золотыми вместо листьев. Каждый раз на дереве вырастает в 10 раз больше золотых, чем было посажено. Сколько раз придется доверчивому деревянному человечку повторять посадку золотых и сбор урожая, чтобы хватило на новый домик для папы Карло? Вводится целое число – стоимость дома. Выведите, сколько раз нужно повторять цикл, чтобы

получить не меньше денег, чем требуется? (10 баллов)

6) Для скорохода и семь миль не крюк. Напишите программу, которая считает пройденные мили и каждый раз выводит сообщение: Пройдена <номер> миля. Кроме случая, когда количество пройденных миль кратно 7. Тут надо вывести: Крюк! и начать считать мили заново. Вводится количество миль. Вывести строки в описанном порядке (10 баллов).

Пример промежуточной аттестации

(базовый уровень)

(Максимальное количество - 40 баллов)

1. Маленький зелёный человечек за добрые дела может отблагодарить скромным серебряным шиллингом, который будет возвращаться к владельцу, а жадине может подарить золотой, который скоро превратится в кусок коры или рассыплется пеплом. Напишите программу для определения подарка от лепрекона. Вводятся строки: **добрый**, **злой** или **Какой подарок?** пока не будет введена пустая строка. Считается количество введенных строк разного типа и на вопрос о подарке выводится ответ. Если строк **добрый** было больше, чем строк **злой**, и последняя строка **добрый**, то подарок **серебряный шиллинг**. Если больше было строк **злой** и такая же последняя введенная, то подарок – **золотой**. Если вопрос задан при неопределенном значении подарка, выводится: **Ах, не знаю!** и программа завершает работу. После каждого вопроса строки считаются заново (8 баллов).

2. Для скорохода и семь миль не крюк. Напишите программу, которая считает пройденные мили и каждый раз выводит сообщение: **Пройдена <номер> миля**. Кроме случая, когда количество пройденных миль кратно 7. Тут надо вывести: **Крюк!** и начать считать мили заново. Вводится количество миль. Вывести строки в описанном порядке (8 баллов).

3. Выведите **n** строк по **n** чисел от 1 до n^2 через символ табуляции (8 баллов).

4. Напишите программу, которая находит сумму степеней всех чисел от 1 до n , степень каждого числа i находится как сумма чисел той же четности, что и само число i , от 1 до i : $11+22+31+3+42+4+51+3+5+...$ (8 баллов)

5. Гусь стоит 25 пиастров, утка – 10, а курица 7. Всего нужно купить n птиц, а в наличии есть m пиастров. Сколько каких птиц можно купить на эти деньги? Вводится два числа – n и m . В произвольном порядке вывести все возможные варианты количества птиц, которых можно купить на все деньги, если их общее количество равно заданному. В каждой строке сначала количество гусей, затем уток и потом кур (8 баллов).

Пример промежуточной аттестации

(продвинутый уровень)

(Максимальное количество - 40 баллов)

1. Тема: SQL базовые операции (4 балла)

Какой SQL-оператор используется для получения (выборки) данных из таблицы базы данных?

- A) INSERT
- B) UPDATE
- C) SELECT
- D) DELETE

2. Тема: SQL добавление данных (4 балла)

Какой SQL-оператор предназначен для добавления новых строк в таблицу базы данных?

- A) ADD
- B) INSERT
- C) CREATE
- D) APPEND

3. Тема: SQL обновление данных (4 балла)

Какой SQL-оператор используется для изменения существующих записей в таблице?

- A) MODIFY
- B) CHANGE
- C) UPDATE
- D) ALTER

4. Тема: SQL удаление данных (4 балла)

Какой SQL-оператор удаляет строки из таблицы базы данных?

- A) REMOVE
- B) DROP
- C) CLEAR
- D) DELETE

5. Тема: Форматы хранения данных (4 балла)

Какой формат файлов используется для хранения табличных данных, где значения разделены запятыми?

- A) XML
- B) CSV
- C) JSON
- D) YAML

6. Тема: Системы контроля версий (4 балла)

Что такое «коммит» (commit) в системе контроля версий Git?

- A) Удаление файла из репозитория
- B) Сохранение изменений в истории версий
- C) Создание новой ветки разработки
- D) Объединение двух веток

7. Тема: HTTP запросы (4 балла)

Какой тип HTTP-запроса обычно используется для получения данных от сервера без их изменения?

- A) POST
- B) PUT
- C) GET
- D) DELETE

8. Тема: Сериализация данных (4 балла)

Какой формат сериализации данных основан на парах «ключ-значение» и широко используется в WEB-API?

- A) XML
- B) CSV
- C) JSON
- D) HTML

9. Тема: API (4 балла)

Что такое API в контексте веб-разработки?

- A) Графический интерфейс пользователя
- B) Интерфейс для взаимодействия между программами
- C) База данных для хранения пользователей
- D) Веб-сервер для хостинга сайтов

10. Тема: Библиотека requests (4 балла)

Какая Python-библиотека наиболее популярна для отправки HTTP-запросов к API?

- A) http_client
- B) urllib
- C) requests
- D) web_connect

Аннотация

Дополнительная общеобразовательная общеразвивающая программа «Программирование на Python» предназначена для изучения программирования на языке Python. Программа является каркасной, что позволяет использовать в процессе обучения программное обеспечение в зависимости от материально-технических возможностей учебных аудиторий, а также знаний и навыков преподавателя.

Дополнительная общеобразовательная общеразвивающая программа «Программирование на Python» имеет техническую направленность, ориентирована на изучение различных аспектов алгоритмизации задач и разработки программ, синтаксических конструкций языка Python, применение языка программирования Python в различных областях IT-сферы.

Цель программы: формирование у обучающихся научно-технических компетенций и практических навыков в области программирования на языке Python.

Дополнительная общеобразовательная общеразвивающая программа «Программирование на Python» предназначена для обучающихся в возрасте 12–17 лет, проявляющих интерес к информационным технологиям и областям знаний технической направленности.

Программа «Программирование на Python» состоит из трех модулей: первый модуль является стартовым уровнем сложности, второй – базовым, третий – продвинутым. Срок освоения программы определяется содержанием программы и составляет 3 года (144 часа в год).