


МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Министерство образования и науки Удмуртской Республики
Муниципальное образование "Муниципальный округ Глазовский
район Удмуртской Республики"
МОУ "Ключевская СОШ"

РАССМОТРЕНО

На педагогическом
совете №1 от 29.08.2024



УТВЕРЖДЕНО
Директор МОУ «Ключевская СОШ»


Д.Г. Главатских
Приказ №78-од от 31.08.2024



Дополнительная общеобразовательная общеразвивающая программа
«Основы программирования на языке Python»

Адресат программы:
обучающиеся 12-17 лет

Срок реализации: 1 год

Составитель: Горбушин А.Г.

Удм.Ключи, 2024г.

В настоящее время мы переживаем большие изменения в развитии современного общества. В современную жизнь человека всё больше внедряются компьютеры и информационные технологии. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причём зачастую не на пользовательском уровне, а на уровне начинающего программиста.

В обязательном школьном курсе информатики программирование нередко представлено лишь на элементарном уровне, на это выделяется недостаточное количество часов. Лишь немногие школы могут себе позволить преподавать программирование на достойном уровне. Следствием этого является формальное восприятие учащимися основ современного программирования и неумение применять полученные знания на практике.

Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке. Нами был выбран язык Python. Его синтаксис достаточно прост и интуитивно понятен, что понижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических аспектах программирования, а не на выучивании тонкостей синтаксиса. При этом Python является очень востребованным языком; он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до машинного обучения.

Научившись программировать на языке Python, учащиеся получают мощный и удобный инструмент для решения как учебных, так и прикладных задач. Вместе с тем чистота и ясность его конструкций позволит учащимся потом с лёгкостью выучить любой другой язык программирования.

Знания и умения, приобретённые в результате освоения курса, могут быть использованы обучающимися при сдаче ОГЭ и ЕГЭ, при участии в олимпиадах по программированию, при решении задач по физике, химии, биологии, лингвистике и другим наукам, а также они являются фундаментом для дальнейшего совершенствования мастерства программирования.

В основу курса «Основы программирования на языке Python» заложены принципы модульности и практической направленности, что обеспечит вариативность обучения. Содержание учебных модулей направлено на:

- детальное изучение алгоритмизации;
- реализацию межпредметных связей;
- организацию проектной и исследовательской деятельности обучающихся.

Важным аспектом программы является самостоятельная работа над заданиями: школьники учатся решать задачи без помощи преподавателя. Для этого в содержании курса фигурируют задания, в которых:

- для решения задачи необходимо найти какую-то информацию в сети Интернет;
- может потребоваться устранение ошибки, которую не так просто быстро обнаружить;
- условие сформулировано недостаточно прозрачно и ученику необходимо самостоятельно доформулировать его (или задать правильные вопросы преподавателю).

Информационные материалы и литература. Рабочая программа «Основы программирования на языке Python» на основе центра «Точка роста» составлена на основе Федерального государственного образовательного стандарта основного общего образования (Приказ Минобрнауки России от 17 декабря 2010 г. № 1897 «Об утверждении федерального образовательного стандарта основного общего образования» (и изменениями, внесенными приказом Минобрнауки России от 31 декабря 2015 г. № 1577)), Федеральный закон от 29.12.2012г. №273-ФЗ «Об образовании в Российской Федерации», статей: 2 (п.9, п.14, п.15, п.22), 12 (п.1ч.4; ч.5), 13 (ч.1,2), 28 (п.6 ч.3), 33 (ч.2), 55 (п.2 ч.1), 58 (ч.1), 75 (ч.1, ч.4); Приказ Министерства просвещения РФ от 09.11.2018г. №196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительной общеобразовательным программам»; СанПин 2.4.4.3172-14, утверждённый Постановлением

Главного государственного санитарного врача РФ от 04.07.2014г. №41; Концепция развития дополнительного образования детей (Распоряжение Правительства Российской Федерации от 4 сентября 2014 года № 1726-р); Концепция развития и воспитания в Российской Федерации на период до 2025 года (утв. Распоряжением Правительства РФ от 28.05.2015г. №996-р); Концепция развития дополнительного образования детей (утверждена распоряжением Правительства РФ от 04.09.2014 №1726-р); ООП и учебного плана МОУ Ключевской СОШ.

Программа составлена с соответствии с требованиями ФГОС среднего общего образования, в состав учебно методического комплекса входят:

- ресурсы интернет по теме «Основы программирования на Python» и «Программирование», Яндекс.Лицей
- методическое пособие для учителя;
- сетевая методическая служба авторского коллектива для педагогов на сайте издательства (<http://lbz.ru/>);
- справочник по программированию на языке Python 3;
- материалы для практических работ;
- коллекция олимпиадных заданий по программированию.

Актуальность и педагогическая целесообразность программы.

Python – это язык программирования общего назначения, распространяемый с открытыми исходными текстами. Он оптимизирован для создания качественного программного обеспечения. Язык Python используется сотнями тысяч разработчиков по всему миру в таких областях, как создание веб-сценариев, системное программирование, создание пользовательских интерфейсов, настройка программных продуктов под пользователя, численное программирование и в других. Как считают многие, один из самых используемых языков программирования в мире.

Актуальность программы состоит в том, что активизация познавательного процесса позволяет учащимся более полно выражать свой творческий потенциал и реализовывать собственные идеи в изучаемой области знаний, создаёт предпосылки по применению информационных компетенций в других учебных курсах, а также способствует возникновению дальней мотивации, направленной на освоение профессий, связанных с разработкой программного обеспечения.

Курс служит средством внутрипрофильной специализации в области новых информационных технологий, что способствует созданию дополнительных условий для проявления индивидуальных образовательных интересов учащихся, их дальнейшей профессиональной ориентации.

Программа «Основы программирования на языке Python»- это дополнительная образовательная программа для школьников, ориентированных на развитие своих навыков программирования, владение которыми поможет в будущем изучить сквозные цифровые технологии (большие данные, промышленный интернет, искусственный интеллект и др.). В рамках данной программы учащиеся познакомятся с языком программирования Python. Python – это лаконичный язык программирования, поэтому программы на нём короткие и понятные. Функционал Python расширяется модулями и библиотеками, это позволит при дальнейшем его изучении стать специалистом по машинному обучению и написать свою нейросеть. Таким образом, изучение языка программирования Python позволит подготовить кадры, способные изучить сквозные цифровые технологии. Программа является дополнением к школьному курсу информатики, развивающей компетенции в области программирования и алгоритмизации и расширяющей кругозор в вопросах командной разработки. Знание основ программирования позволит слушателям курса сформировать для себя базовые знания о больших данных и искусственном интеллекте.

Данная дополнительная общеразвивающая программа направлена на развитие логического и пространственного мышления слушателя, способствует раскрытию творческого потенциала личности, формированию усидчивости и трудолюбия,

приобретению практических умений и навыков в области компьютерных технологий, способствует интеллектуальному развитию ребенка. Целесообразность программы выражена в подборе интерактивных и практикоориентированных форм занятий, способствующих формированию основных компетенций (информационных, коммуникативных, компетенций личного развития и др.).

Направленность дополнительной общеразвивающей программы – техническая.

Основы инженерного и технического образования – а именно такие специалисты сегодня, да и в ближайшем будущем будут остро нужны стране – закладываются именно в школе. Программа «Основы программирования на языке Python» соответствует Концепции инженерно-технической подготовки школьников. Концепция инженерно-технической подготовки школьников разработана на основании поручения Президента Российской Федерации от 4 мая 2016г с учетом Стратегии научно-технологического развития Российской Федерации, утвержденной Указом Президента Российской Федерации от 1 декабря 2016г. №642, Национальной технологической инициативы, (постановление Правительства Российской Федерации от 18 апреля 2016г. №317 «О реализации Национальной технологической инициативы») и Программы «Цифровая экономика Российской Федерации», утвержденной распоряжением Правительства Российской Федерации от 28 июля 2017г. №1632-р. Среди задач, сформулированных в Концепции: – изучение элементов как традиционных, так и наиболее перспективных технологических направлений; – формирование у обучающихся культуры проектной и исследовательской деятельности; – формирование ключевых навыков в сфере информационных и коммуникационных технологий; – создание системы выявления, оценивания и продвижения обучающихся, обладающих высокой мотивацией и способностями в сфере материального и социального конструирования, включая инженерно-технологическое направление и ИКТ.

Требуемый уровень подготовки обучающихся - начинающие и опытные пользователи персонального компьютера.

Адресат программы – обучающихся 12-17 лет.

Срок освоения программы – 1 год.

Форма обучения – очная.

Режим занятий – 1 раз в неделю по 2 учебных часа, перерыв между занятиями 10 минут). Объем 72 часа в год.

Цель: способствовать формированию творческой личности, обладающей информационными компетенциями, владеющей базовыми понятиями теории алгоритмов, умеющей разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

Задачи:

Образовательные:

1. обучить языку программирования Python и созданию программ на его основе;
2. научить создавать прикладное программное обеспечение;
3. расширить кругозор обучающихся в области программирования;
4. научить дизайнерскому оформлению созданного ПО.

Развивающие:

1. развивать память и внимание, познавательную и творческую активность;
2. развивать творческие способности, эстетическое и эргономическое восприятие объектов труда;
3. развивать логическое мышление.

Воспитательные:

1. прививать интерес к активному творческому самовыражению, культуре труда;
2. воспитывать упорство в достижении желаемого результата;
3. воспитывать эстетический вкус;
4. воспитывать коммуникативные навыки.

Психологические особенности. Программа ориентирована на детей в возрасте 12-17 лет. Отбор в группы по собеседованию, так как, предлагаемая программа предназначена для обучающихся, владеющих начальными навыками работы на персональном компьютере. Подростковый возраст это весьма сложный период в жизни ребенка. У него возникает представление о себе уже не как о ребенке, он стремится быть и считаться взрослым. Подросток становится активным, восприимчивым к усвоению норм, ценностей и способов поведения, которые характерны для взрослого самостоятельного человека. Период подросткового возраста характерен тем, что формируется морально нравственные и социальные установки личности обучающегося. Подросток стремится к активному общению со сверстниками, и через это общение он познает самого себя. У него возникают потребности, которые он должен удовлетворять только сам (потребность в общении со сверстниками, дружбе и любви). Характерной чертой этого возраста является любознательность, пылкость ума, стремление к познанию и информации, подросток стремится овладеть как можно большим количеством знаний, но не обращая порой внимания, что знания надо систематизировать.

Данная программа позволяет школьникам выстроить индивидуальную образовательную траекторию технологического, естественно-математического и универсального профилей средней (полной) общеобразовательной школы. Содержание программы построено таким образом, чтобы обеспечить возможность его усвоения учащимися, имеющими разную стартовую подготовку.

Новизна данной образовательной программы

Программа курса:

- обеспечивает знакомство с фундаментальными понятиями алгоритмизации и программирования на доступном уровне;
- имеет практическую направленность с ориентацией на реальные потребности, соответствующие возрасту ученика;
- охватывает как алгоритмическое направление, так и вопросы практического использования полученных знаний при решении задач из различных областей знаний;
- ориентирована на существующий парк вычислительной техники и дополнительные ограничения;
- допускает возможность варьирования в зависимости от уровня подготовки и интеллектуального уровня учащихся (как группового, так и индивидуального);
- предусматривает возможность индивидуальной работы с учащимися.

Во время обучения, учащиеся получают знания, умения и навыки, которые в дальнейшем позволят им самим планировать и осуществлять трудовую деятельность.

Программа направлена на воспитание современных детей как творчески активных и технически грамотных начинающих инженеров, способствует возрождению интереса молодежи к технике, в воспитании культуры жизненного и профессионального самоопределения.

Практическая значимость. Данная программа уникальна по своим возможностям и направлена на знакомство с современными технологиями и стимулированию интереса учащихся к технологиям программирования.

Практическая значимость курса заключается в том, что он способствует более успешному овладению знаниями и умениями по направлению «Программирование» через развитие самостоятельности обучающихся и оптимизацию средств и методов обучения.

Педагогическая целесообразность данной программы:

- взаимодействие педагога с ребенком на равных;
- использование на занятиях доступных для детей понятий и терминов, следование принципу «от простого к сложному»;
- учет разного уровня подготовки детей, опора на имеющийся у обучающихся опыт;
- системность, последовательность и доступность излагаемого материала, изучение нового материала опирается на ранее приобретенные знания;

- приоритет практической деятельности;
- развитие в учащихся самостоятельности, творчества и изобретательности является одним из основных приоритетов данной программы.

В рамках программы «Основы программирования на языке Python» изучение основ программирования на языке Python – это не столько средство подготовки к будущей профессиональной деятельности, сколько формирование новых общеинтеллектуальных умений и навыков: разделение задачи на этапы решения, построение алгоритма и др. Особую роль программирование служит для формирования мыслительных и психических процессов учащихся (внимание, память, логика), освоения приёмов умственных действий, самостоятельного нахождения и составления алгоритмов решения задач, умения строить модели, чётко и лаконично реализовывать этапы решения задач. Использование этих возможностей для формирования предметных и метапредметных результатов учащихся особенно важно, т.к. именно они активизирует процесс индивидуально-личностного становления учащихся.

Отличительные особенности программы.

На занятиях учащиеся познакомятся с теоретическими аспектами и синтаксисом языка, а также обучатся практическим навыкам программирования в среде Python. Занятия начинаются с практического знакомства со средой программирования Python, далее идет непосредственное изучение синтаксических конструкций языка и отработка навыков применения элементов программирования при решении задач и создании игр. Каждая новая тема завершается практическими задачами, способствующими овладению методики программирования и изучению языка Python. Обучения закрепляются полученные знания и навыки программирования, осваиваются новые методы, способы решения задач, рассматриваются более сложные задачи, новые технологии программирования.

Ключевой особенностью является его направленность на формирование у учащихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и его реализации с помощью средств программирования.

Отличительной особенностью программы является её направленность на формирование у воспитанников потребности в инженерном образовании и научном исследовании. Инновационной является авторская методика, позволяющая сочетать изучение достаточно сложного языка программирования с исследовательской деятельностью и разработкой проектов самых разнообразных по содержанию, позволяющая ребенку в конце курса обучения почувствовать себя настоящим профессионалом.

Технологии, используемые в образовательном процессе:

Технологии традиционного обучения для освоения минимума содержания образования в соответствии с требованиями стандартов; технологии, построенные на основе объяснительно-иллюстративного способа обучения. В основе – информирование, просвещение обучающихся и организация их репродуктивных действий с целью выработки у школьников общеучебных умений и навыков.

Технологии компьютерных практикумов.

Технологии реализации межпредметных связей в образовательном процессе.

Технологии дифференцированного обучения для освоения учебного материала обучающимися, различающимися по уровню обучаемости, повышения познавательного интереса.

Технология проблемного обучения с целью развития творческих способностей обучающихся, их интеллектуального потенциала, познавательных возможностей. Обучение ориентировано на самостоятельный поиск результата, самостоятельное добывание знаний, творческое, интеллектуально-познавательное усвоение учениками заданного предметного материала.

Личностно-ориентированные технологии обучения, способ организации обучения, в процессе которого обеспечивается всемерный учет возможностей и способностей обучаемых и создаются необходимые условия для развития их индивидуальных способностей.

Информационно-коммуникационные технологии.

Технология коллективных методов обучения (работа в парах постоянного и сменного состава).

СОДЕРЖАНИЕ ПРОГРАММЫ

Образовательная программа «Основы программирования на языке Python» состоит из 8 разделов:

Раздел 1. Знакомство с языком Python (3 часа)

Теория: Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python. Что такое программа. Первая программа. Структура программы на языке Python. Комментарии.

Практическая работа 1.1. Установка программы Python Практическая работа 1.2. Режимы работы с Python Тест № 1. Знакомство с языком Python

Учащиеся должны знать / понимать:

- понятие программы;
- структура программы на Python;
- режимы работы с Python.

Учащиеся должны уметь:

- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе.

Раздел 2. Переменные и выражения (6 часов)

Теория: Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова.

Выражения. Операции. Порядок выполнения операций. Математические функции. Композиция.

Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных. Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.

Практическая работа 2.1. Работа со справочной системой Практическая работа 2.2. Переменные

Практическая работа 2.3. Выражения

Практическая работа 2.5. Задачи на элементарные действия с числами Тест № 2.

Выражения и операции.

Учащиеся должны знать / понимать:

- общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;

Раздел 3. Условные предложения (14 часов)

Теория: Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python.

Практическая работа 3.1. Логические выражения Практическая работа 3.2. "Условный оператор" Практическая работа 3.3. Множественное ветвление Практическая работа 3.4. "Условные операторы"

Самостоятельная работа № 1. Решение задач по теме "Условные операторы". Зачетная работа № 1. "Составление программ с ветвлением".

Тест № 3. "Условные операторы".

Учащиеся должны знать / понимать:

- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- Учащиеся должны уметь:
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов.

Раздел 4. Циклы (18 часов)

Теория: Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов.

Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

Практическая работа 4.1. "Числа Фибоначчи" Практическая работа 4.2. Решение задачи с циклом for.

Практическая работа 4.3. Реализация циклических алгоритмов Практическая работа 4.4. Случайные числа

Практическая работа 4.5. Решение задач с циклом. Самостоятельная работа № 2 "Составление программ с циклом"

Тест № 4. Циклы

Творческая работа № 1. "Циклы"

Учащиеся должны знать / понимать:

- циклы с условием и их виды;
- правила записи циклов условием;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.

Учащиеся должны уметь:

- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использовать цикл с параметром для решения поставленной задачи.

Раздел 5. Функции (12 часов)

Теория: Создание функций. Параметры и аргументы. Локальные и глобальные переменные.

Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Примеры решения задач с использованием функций.

Рекурсивные функции. Вычисление факториала. Числа Фибоначчи. Практическая работа 5.1. Создание функций

Практическая работа 5.2. Локальные переменные

Практическая работа 5.3. Решение задач с использованием функций Практическая работа 5.4. Рекурсивные функции

Самостоятельная работа № 3 по теме "Функции" Тест № 5. Функции

Учащиеся должны знать / понимать:

- понятие функции;
- способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.

Учащиеся должны уметь:

- создавать и использовать функции;
- использовать механизм параметров для передачи значений.

Раздел 6. Строки - последовательности символов (3 часа)

Теория: Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки.

Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.

Практическая работа 6.1. Строки

Практическая работа 6.2. Решение задач со строками.

Учащиеся должны знать / понимать:

- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками.

Учащиеся должны уметь:

- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке.

Раздел 7. Сложные типы данных (8 часов)

Теория: Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range. Списки: примеры решения задач.

Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения

Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы.

Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.

Практическая работа 7.1. Списки.

Практическая работа 7.2. Решение задач со списками. Тест № 7. Списки

Учащиеся должны знать / понимать:

- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.

Учащиеся должны уметь:

- описывать списки;
- вводить элементы списка;
- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;

- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.

Раздел 8. Стиль программирования и отладка программ (8 часов)

Стиль программирования. Отладка программ.

Зачет по курсу «Основы программирование на языке Python»

Учащиеся должны знать / понимать:

- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ.

Учащиеся должны уметь:

- определять вид ошибок и находить ошибки в программе.
- выполнять тестирование и отладку программ.

Методы и формы обучения: в разновозрастных группах используют фронтальные, групповые и индивидуальные формы организации образовательного процесса, которые позволяют разным образом формировать взаимоотношения педагога с детьми и детей между собой. Наиболее эффективным, есть сочетание разных форм работы (коллективная работа, работа с подгруппой и индивидуальные занятия). Более общие образовательные задачи лучше решать на фронтальных занятиях, а конкретные (сообщение нового материала, закрепление, расширение и уточнение знаний) - на занятиях с одной подгруппой. При организации учебно-воспитательного процесса необходимо учитывать индивидуальные, возрастные особенности детей. При проведении комплексных занятий в разновозрастной группе надо следить за тем, чтобы деятельность детей одной подгруппы не отвлекала детей другой подгруппы. Общие занятия целесообразно проводить при условии одинаковой или близкой темы для детей всех возрастных подгрупп, учитывая возможности детей и уровень их самостоятельности.

Формы контроля: текущий контроль усвоения материала осуществляться путем практических заданий. Периодически знания и умения по пройденным темам проверяются выполнением тестовых заданий и графических работ. Предметом диагностики и контроля являются внешние образовательные продукты учеников (созданные графические изображения), а также их внутренние личностные качества (освоенные способы деятельности, знания, умения), которые относятся к целям и задачам программы.

Основой для оценивания деятельности учеников являются результаты анализа его продукции и деятельности по ее созданию. Оценка имеет различные способы выражения - устные суждения педагога, письменные качественные характеристики.

Оценке подлежит в первую очередь уровень достижения учеником минимально необходимых результатов, обозначенных в целях и задачах программы. Оцениванию подлежат также те направления и результаты деятельности учеников, которые определены в рабочей программе педагога и в индивидуальных образовательных программах учеников.

Ученик выступает полноправным субъектом оценивания. Одна из задач педагога — обучение детей навыкам самооценки. С этой целью учитель выделяет и поясняет критерии оценки, учит детей формулировать эти критерии в зависимости от поставленных целей и особенностей образовательного продукта.

Проверка достигаемых учениками образовательных результатов производится в следующих формах:

✓ текущий рефлексивный самоанализ, контроль и самооценка учащимися выполняемых заданий - оценка промежуточных достижений используется как инструмент положительной мотивации, для своевременной коррекции деятельности учащихся и учителя; осуществляется по результатам выполнения учащимися практических заданий на каждом занятии;

- ✓ взаимнооценка учащимися работ друг друга или работ, выполненных в группах;
- ✓ текущая диагностика и оценка учителем деятельности школьников;
- ✓ итоговый контроль проводится в конце программы в форме творческих работ (индивидуальных или групповых);

На основе творческих работ проводятся конкурсы и выставки, формируются «портфолио» учащихся. Данный тип контроля предполагает комплексную проверку образовательных результатов по всем заявленным целям и направлениям программы. Формой итоговой оценки каждого ученика выступает образовательная характеристика, в которой указывается уровень освоения им каждой из целей программы и каждого из направлений индивидуальной программы ученика.

Основной тип занятий — комбинированный, сочетающий в себе элементы теории и практики. Большинство заданий курса выполняется самостоятельно с помощью персонального компьютера и необходимых программных средств.

Единицей учебного процесса является блок уроков (модуль). Каждый такой блок охватывает отдельную информационную технологию или её часть. Внутри блоков разбивка по времени изучения производится учителем самостоятельно, но с учётом рекомендованного календарно-тематического плана. С учётом регулярного повторения ранее изученных тем темп изучения отдельных разделов блока определяется субъективными и объективными факторами.

Каждая тема курса начинается с постановки задачи — характеристики предметной области или конкретной программы на языке Python, которую предстоит изучить. С этой целью учитель проводит демонстрацию презентации или показывает саму программу, а также готовые работы, выполненные в ней. Закрепление знаний проводится с помощью практики отработки умений самостоятельно решать поставленные задачи, соответствующих минимальному уровню планируемых результатов обучения.

Основные задания являются обязательными для выполнения всеми обучающимися в классе. Задания выполняются на компьютере с использованием интегрированной среды разработки. При этом ученики не только формируют новые теоретические и практические знания, но и приобретают новые технологические навыки.

В каждую тему курса интегрированы вопросы, связанные с теоретическими основами изучаемого раздела. Это позволяет учащимся получить более глубокое представление о принципах работы с информацией и способах автоматизации этой деятельности.

Методика обучения ориентирована на индивидуальный подход. Для того чтобы каждый ученик получил наилучший результат обучения, программой предусмотрены дополнительные задания для самостоятельной работы на домашнем компьютере. Такая форма организации обучения стимулирует интерес ученика к предмету, активность и самостоятельность учащихся, способствует объективному контролю глубины и широты знаний, повышению качества усвоения материала обучающимися, позволяет педагогу получить объективную оценку выбранной им тактики и стратегии работы, методики индивидуального обучения и обучения в группе, выбора предметного содержания.

Для самостоятельной работы используются разные по уровню сложности задания, которые носят репродуктивный и творческий характер. Количество таких заданий в работе может варьироваться.

В ходе обучения проводится промежуточное тестирование по темам для определения уровня знаний учеников. Выполнение контрольных заданий способствует активизации учебно-познавательной деятельности и ведёт к закреплению знаний, а также служит индикатором успешности образовательного процесса.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ (ЛИЧНОСТНЫЕ, МЕТАПРЕДМЕТНЫЕ, ПРЕДМЕТНЫЕ)

В результате изучения дисциплины получают дальнейшее развитие личностные регулятивные, коммуникативные и познавательные-универсальные учебные действия, учебная (общая и предметная) и общепользовательская ИКТ–компетентность обучающихся, составляющая психолого-педагогическую, инструментальную основу формирования способности и готовности к освоению систематических знаний, к их самостоятельному пополнению, переносу и интеграции, способности к сотрудничеству и коммуникации, решению лично и социально значимых проблем и воплощению решений в практику, способности к самоорганизации, саморегуляции и рефлексии.

В ходе изучения данного курса в основном формируются и получают развитие следующие метапредметные результаты:

- умение самостоятельно планировать пути достижения цели, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения задач;
- умение соотносить свои действия с планируемыми результатами, осуществлять контроль всей деятельности в процессе достижения результата, корректировать свои действия в соответствии с изменяющейся ситуацией;
- умение оценивать правильность выполнения учебной задачи, собственные возможности ее решения;
- владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
- умение организовывать сотрудничество и совместную деятельность с учителем и сверстниками; работать индивидуально в группе: находить общее решение и разрешать конфликты на основе согласования позиций и учета интересов; формулировать, аргументировать и отстаивать свое мнение;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий.

Вместе с тем, вносится существенный вклад в развитие личностных результатов:

- формирование ответственного отношения к обучению, готовности и способности обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками, взрослыми в процессе образовательной, общественно-полезной, учебно-исследовательской, творческой и других видах деятельности.

В части развития предметных результатов наибольшее влияние изучение курса оказывает на:

- формирование знаний, умений и навыков при решении задач информатики и программирования разных видов;
- приобретение опыта использования информационных технологий в индивидуальной и коллективной учебной и познавательной, в том числе проектной деятельности.
- формирование информационной и алгоритмической культуры; формирование представления о компьютере как универсальном устройстве обработки информации; развитие основных навыков и умений использования компьютерных устройств;
- формирование представления об основных изучаемых понятиях: информация, алгоритм, модель – и их свойства;
- развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм для конкретного исполнителя; формирование знаний об алгоритмических конструкциях, логических значениях и операциях; знакомство с одним из языков программирования и основными алгоритмическими структурами – линейной, условной и циклической;
- владение системой базовых знаний, отражающих вклад информатики в формирование современной научной картины мира;

- овладение понятием сложности алгоритма, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки;
- владение универсальным языком программирования высокого уровня Python, представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции;
- владение навыками и опытом разработки программ в среде программирования Python, включая тестирование и отладку программ; владение элементарными навыками формализации прикладной задачи и документирования программ;
- формирование умения работать с библиотеками программ; получение опыта использования компьютерных средств представления и анализа данных.

Hard компетенции:

будут знать:

- принципы программирования на языке Python;
- основные типы алгоритмов;
- иметь представление о структуре программы, основы программирования на языках высокого уровня;
- базовые алгоритмические конструкции;
- содержание этапов разработки программы: алгоритмизация-кодирование-отладка-тестирование;
- дополнительные возможности языка Python для выражения различных алгоритмических ситуаций;
- алгоритмы и программы на языке Python решения простых, сложных и нестандартных задач в математической области;
- основы разработки простых игр в системе программирования Python.
- основы дизайнерского оформления созданных программ.

будут уметь:

- производить чтение и запись программ на языке Python;
- запускать и отлаживать программу.
- записывать алгоритмические структуры на языке программирования Python;
- использовать Python для решения задач из области математики, физики;
- строить алгоритмы методом последовательного уточнения (сверху вниз), изображать эти алгоритмы в виде блок-схем;
- использовать основные алгоритмические приемы при решении математических задач;
- решать простые, сложные и нестандартные задачи;
- создавать простые игры;
- анализировать текст чужих программ, находить в них неточности, оптимизировать алгоритм, создавать собственные варианты решения.

Soft компетенции:

- формирование ответственного отношения к учению, готовности и способности учащихся к саморазвитию и самообразованию;
- развитие самостоятельности, личной ответственности за свои поступки;
- мотивация детей к познанию, творчеству, труду;
- формирование осознанного, уважительного и доброжелательного отношения к другому человеку;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками в процессе разных видов деятельности.
- формирование умения самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности;
- формирование умения самостоятельно планировать пути достижения целей,

- осознанно выбирать наиболее эффективные способы решения учебных и познавательных задач;
- формирование умения понимать причины успеха/неуспеха учебной деятельности;
- овладение различными способами поиска информации в соответствии с поставленными задачами;
- готовность слушать собеседника и вести диалог; излагать свое мнение и аргументировать свою точку зрения;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий.

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№ п/п	Название раздела, темы	Количество часов			Форма промежуточной (итоговой) аттестации
		Всего	Теория	Практика	
1.	Раздел 1. Знакомство с языком Python (3 часа)	3	1	2	Опрос Тестирование, решение практических задач
1.1	Инструктаж по технике безопасности в компьютерном классе Занятие 1. Общие сведения о языке Практическая работа: Установка программы Python	1	1		
1.2.	Занятие 2. Режимы работы Практическая работа: Режимы работы с Python	1		1	
1.3	Тест № 1. Знакомство с языком Python	1		1	
2.	Раздел 2. Переменные и выражения (6 часов)	6	2	4	Тестирование, решение практических задач
2.1	Занятие 3. Переменные Практическая работа: Работа со справочной системой	1		1	
2.2	Практическая работа: Переменные	1		1	
2.3	Занятие 4. Выражения Практическая работа: Выражения	1		1	
2.4	Занятие 5. Ввод и вывод	1	1		
2.5	Занятие 6. Задачи на элементарные действия с числами Практическая работа 2.5. Задачи на элементарные действия с числами	1	1	1	
2.6	Тест № 2. Выражения и операции.	1		1	
3	Раздел 3. Условные предложения (14 часов)	14	4	10	Тестирование, Решение практических

3.1	Занятие 7. Логические выражения и операторы. Практическая работа: Логические выражения	2	1	1	задач
3.2	Занятие 8. Условный оператор Практическая работа: «Условный оператор»	2	1	1	
3.3	Занятие 9. Множественное ветвление Практическая работа: Множественное ветвление	2	1	1	
3.4	Занятие 10. Реализация ветвления в языке Python. Практическая работа: «Условные операторы»	2	1	1	
3.5	Самостоятельная работа № 1 по теме «Условные операторы».	2		2	
3.6	Занятие 11. Зачетная работа № 1. "Составление программ с ветвлением".	2		2	
3.7	Тест № 3. "Условные операторы".	2		2	
4	Раздел 4. Циклы (18 часов)	18	6	12	Тестирование, решение практических задач, творческая работа
4.1.	Занятие 12. Оператор цикла с условием Практическая работа "Числа Фибоначчи"	2	1	1	
4.2.	Занятие 13. Оператор цикла for Практическая работа Решение задачи с циклом for.	2	1	1	
4.3.	Занятие 14. Вложенные циклы Практическая работа: Реализация циклических алгоритмов	2	1	1	
4.4.	Занятие 15. Случайные числа Практическая работа: Случайные числа	2	1	1	
4.5.	Занятие 16. Примеры решения задач с циклом Практическая работа: Решение задач с циклом.	2	1	1	
4.6	Самостоятельная работа № 2 "Составление программ с циклом"	2	1	1	
4.7	Тест № 4. Циклы	2		2	
4.8	Занятие 17. Творческая работа № 1. "Циклы"	4		4	
5	Раздел 5. Функции (12 часов)	12	4	8	Тестирование, решение практических задач
5.1.	Занятие 18. Создание функций Практическая работа Создание функций	2	1	1	
5.2.	Занятие 19. Локальные	2	1	1	

	переменные Практическая работа Локальные переменные				
5.3.	Занятие 20. Примеры решения задач с использованием функций Практическая работа Решение задач с использованием функций	2	1	1	
5.4	Самостоятельная работа № 3 по теме "Функции"	2		2	
5.5	Занятие 21. Рекурсивные функции Практическая работа 5.4. Рекурсивные функции	2	1	1	
5.6	Тест № 5. Функции	2		2	
6	Раздел 6. Строки - последовательности символов (3 часа)	3	0	3	
6.1.	Занятие 22. Строки Практическая работа: Строки	1		1	Решение практических задач
6.2.	Занятие 23. Срезы строк	1		1	
6.3	Занятие 24. Примеры решения задач со строками Практическая работа: Решение задач со строками.	1		1	
7	Раздел 7. Сложные типы данных (8 часов)	8	3	5	
7.1.	Занятие 25. Списки Практическая работа: Списки	1		1	Тестирование, Решение практических задач
7.2	Занятие 26. Срезы списков	1	1		
7.3	Занятие 27. Списки: примеры решения задач Практическая работа 7.2. Решение задач со списками	1		1	
7.4	Занятие 28. Матрицы	1	1		
7.5	Тест № 7. Списки	1		1	
7.6	Занятие 29. Кортежи	1	1		
7.7	Занятие 30. Введение в словари	1		1	
7.8	Занятие 31. Множества в языке Python	1		1	
8	Раздел 8. Стиль программирования и отладка программ	8	2	6	
8.1	Занятие 32. Стиль программирования	2		2	Решение практических задач, презентация проекта, рефлексия.
8.2	Занятие 33. Отладка программ	2	1	1	
8.3	Занятие 34. Зачет по курсу «Основы программирование на языке Python»	4	1	3	
	ВСЕГО	72	22	50	

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Механизм оценивания результатов реализации программы

Показатели (оцениваемые параметры)	Критерии	Степень выраженности оцениваемого качества	Методы диагностики
1. Теоретическая подготовка детей: 1.1. Теоретические знания (по основным разделам учебного плана программы)	Соответствие теоретических знаний программным требованиям	- низкий уровень (овладели менее чем 50% объема знаний);	Собеседование, тестирование
		- средний уровень (объем освоенных знаний составляет более 50%);	
		- высокий уровень (освоили практически весь объем знаний, предусмотренных программой)	
1.2. Владение специальной терминологией	Осмысленность и правильность использования	- низкий уровень (избегают употреблять специальные термины);	Собеседование, Опрос
		- средний уровень (сочетают специальную терминологию с бытовой);	
		- высокий уровень (термины употребляют осознанно и в полном соответствии с их содержанием)	
2. Практическая подготовка детей: 2.1. Практические умения и навыки, предусмотренные программой (по основным разделам)	Соответствие практических умений и навыков программным требованиям	- низкий уровень (овладели менее чем 50% предусмотренных умений и навыков);	Наблюдения, Итоговые работы
		- средний уровень (объем освоенных умений и навыков составляет более 50%);	
		- высокий уровень (дети овладели практически всеми умениями и навыками, предусмотренными программой)	

2.2. Владение специальным оборудованием и оснащением	Отсутствие затруднений в использовании	- низкий уровень (испытывают серьезные затруднения при работе с оборудованием)	Наблюдение, Итоговые работы
		- средний уровень (работает с помощью педагога)	
		- высокий уровень (работают самостоятельно)	
2.3. Творческие навыки	Креативность в выполнении практических заданий	- низкий (элементарный, выполняют лишь простейшие практические задания)	Наблюдение, Итоговые работы
		- средний (выполняют задания на основе образца)	
		- высокий (выполняют практические задания с элементами творчества)	

Тесты:

1. Деление с остатком: <http://kpolyakov.spb.ru/school/test8a/py20.htm>
2. Ветвления: <http://kpolyakov.spb.ru/school/test8a/py21.htm>
3. Цикл while: <http://kpolyakov.spb.ru/school/test8a/py23.htm>
4. Цикл for: <http://kpolyakov.spb.ru/school/test8a/py24.htm>
5. Строки: <http://kpolyakov.spb.ru/school/test9a/py16a.htm>
6. Списки: <http://kpolyakov.spb.ru/school/test8a/py25.htm>

Практические задания:

1. Вывести на экран рисунок из букв

```

      Ж
     ЖЖЖ
    ЖЖЖЖЖ
   ЖЖЖЖЖЖЖ
  ЖЖ ЖЖ
  ZZZZZ

```

2. Ввести три числа, найти их сумму, произведение и среднее арифметическое.

Пример:

Введите три числа:

4

5

7

$$4+5+7=16$$

$$4*5*7=140$$

$$(4+5+7)/3=5.333333$$

3. Ввести число, обозначающее количество секунд. Вывести то же самое время в часах, минутах и секундах.

Пример:

Введите число секунд: 8325

2 ч. 18 мин. 45 с

4. Ввести последовательно возраст Антона, Бориса и Виктора. Определить, кто из них старше.

Пример:

- Возраст Антона: 15
 Возраст Бориса: 17
 Возраст Виктора: 16
 Ответ: Борис старше всех.
 Пример:
 Возраст Антона: 17
 Возраст Бориса: 17
 Возраст Виктора: 16
 Ответ: Антон и Борис старше Виктора.
5. Напишите программу, которая получает возраст человека (целое число, не превышающее 120) и выводит этот возраст со словом «год», «года» или «лет». Например, «21 год», «22 года», «25 лет».
- Пример:
 Введите возраст: 18
 Вам 18 лет.
 Пример:
 Введите возраст: 21
 Вам 21 год.
 Пример:
 Введите возраст: 22
 Вам 22 года.
6. Напишите программу, которая получает с клавиатуры трёхзначное число и записывает в логическую переменную значение «да» (True), если это число – палиндром, то есть читается одинаково слева направо и справа налево. После этого на экран выводится ответ на вопрос: «Верно ли, что введённое число – палиндром?».
- Пример:
 Введите число: 165
 Ответ: нет.
 Пример:
 Введите число: 656
 Ответ: да.
7. Напишите программу, которая получает с клавиатуры натуральное число и находит наибольшую цифру в его десятичной записи.
- Пример:
 Введите число: 311
 Наибольшая цифра: 3
8. Ввести с клавиатуры два натуральных числа и найти их НОД с помощью алгоритма Евклида.
- Пример:
 Введите два числа:
 21 14
 $\text{НОД}(21,14)=7$
9. Ипполит задумал трёхзначное число, которое при делении на 15 даёт в остатке 11, а при делении на 11 даёт в остатке 9. Напишите программу, которая находит все такие числа.
10. Заполните массив из 10 элементов натуральными числами в обратном порядке, начиная со значения X, введённого с клавиатуры. Последний элемент должен быть равен X, предпоследний равен X-1 и т.д.
11. Ввести с клавиатуры в одну строку фамилию, имя и отчество, разделив их пробелом. Вывести фамилию и инициалы.
- Пример:
 Введите фамилию, имя и отчество:

Иванов Петр Семёнович

П.С. Иванов

12. Напишите программу, которая вычисляет сумму трёх чисел, введенную в форме символьной строки. Все числа целые.

Пример:

Введите выражение:

12+3+45

Ответ: 60

Способы оценивания уровня достижений учащихся

Предметом диагностики и контроля в курсе «Основы программирования на языке Python» являются внешние образовательные продукты учащихся (созданные блок-схемы, программы), а также их внутренние личностные качества (освоенные способы деятельности, знания, умения), которые относятся к целям и задачам курса.

Качество внешней образовательной продукции желательно оценивать по следующим параметрам:

- алгоритм должен быть оптимальным по скорости выполнения и максимально простым в реализации на языке программирования;
- программа должна выполнять поставленные задачи;
- по степени «читаемости кода» (должны быть соблюдены отступы, обязательное наличие комментариев к коду программы и т. д.).
- Проверка достигаемых учащимися результатов производится в следующих формах:
- текущий рефлексивный самоанализ, контроль и самооценка учащимися выполняемых заданий;
- текущая диагностика и оценка педагогом деятельности учащихся;
- итоговая оценка деятельности и образовательной продукции ученика в соответствии с его индивидуальными и возрастными особенностями;
- Итоговый контроль проводится в конце всего курса. Он состоит из тестирования и решения практической задачи, защиты творческого проекта.

Система оценки и критерии результативности освоения программы.

Результат выполнения проверочных работ, текущих работ и зачетных заданий оценивается по 5-балльной шкале:

0 - работа не выполнялась;

1 плохо – работа выполнена не полностью, с большими недочетами, теоретический материал не освоен;

2 удовлетворительно – работа выполнена не полностью, с недочетами, теоретический материал освоен частично;

3 хорошо – работа выполнена полностью, с небольшими недочетами, теоретический материал практически освоен;

4 очень хорошо – работа выполнена в полном соответствии с образцом в указанное время с обращением за помощью к педагогу;

5 отлично – работа выполнена в полном соответствии с образцом в указанное время без помощи педагога.

Итоговый суммарный балл учащегося складывается из баллов:

• за выполнение текущих работ, • за выполнение зачетных проектных заданий,

Итоговая оценка учащегося по Программе (% от максимально возможного итогового балла) отражает результаты учебной работы в течение всего года:

100-70% – высокий уровень освоения программы

69-50% – средний уровень освоения программы

49-30% – низкий уровень освоения программы

Формой итогового контроля усвоения учащимися учебного материала является выполнение зачетного задания и итогового теста.

Примеры оценочных материалов приведены в Приложении.

МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Программа разработана на основе концептуальных положений Стратегии инновационного развития Российской Федерации на период до 2025 года.

В программе применяются следующие методы, позволяющие дать детям первоначальные основы программирования:

Словесные методы позволяют быстро передавать детям информацию, ставить перед ними учебную задачу, указывать пути ее решения. Это лекции, беседы, дискуссии. Словесные методы и приемы сочетаются с наглядными, игровыми, практическими методами, делая последние более результативными.

Наглядные методы: Наглядность оживляет процесс обучения, способствует побуждению у детей интереса к занятию. В качестве наглядности используются презентации, мультимедийные учебные пособия, флеш-анимации, видеоролики, веб-кейсы и другие интернет – технологии, демонстрационный раздаточный материал, карточки.

Практические методы: Нельзя научить детей, только показывая и рассказывая, не предлагая самим детям каких - либо действий. Ребенок овладевает опытом только тогда, когда сам участвует в практической деятельности. практической работы учащихся за компьютером. При решении задач используется семинарское занятие

Игровые методы: позволяют осуществлять учебные задачи в атмосфере легкости и заинтересованности, активности детей. Используются игровые задания, создание игр на компьютере, метод интерактивной игры, дидактические игры, и упражнения, выполнив которые ребенок легко может усвоить правила поведения, технику безопасности, гимнастику для глаз. Формы обучения по программе:

Фронтальная - подача учебного материала всему коллективу детей, на этих занятиях важен «эффект эмоционального воздействия и сопереживания», что приводит к повышению умственной активности, побуждает ребенка к самовыражению (интегрированные и итоговые занятия, интеллектуальные игры).

Индивидуальная - используется при возникновении затруднения, не уменьшая активности детей и содействуя выработке навыков самостоятельной работы. В индивидуальных занятиях нуждаются дети с явно выраженными способностями к той или иной деятельности, дети с доминирующим познавательным интересом.

Подгрупповая - предоставляется возможность самостоятельно построить свою деятельность на основе принципа индивидуализации и сознательности и активности, ощутить помощь со стороны друг друга, учесть возможности каждого на конкретном этапе деятельности. Всё это способствует более быстрому выполнению задания.

Технологии:

1. Технология проблемного диалога. Учащимся не только сообщаются готовые знания, но и организуется такая их деятельность, в процессе которой они сами делают «открытия», узнают что-то новое и используют полученные знания и умения для решения жизненных задач.

2. Технология коллективного взаимообучения («организованный диалог», «сочетательный диалог», «коллективный способ обучения (КСО), «работа учащихся в парах сменного состава») позволяет плодотворно развивать у обучаемых самостоятельность и коммуникативные умения.

3. Игровая технология. Игровая форма в образовательном процессе создаётся при помощи игровых приёмов и ситуаций, выступающих как средство побуждения к деятельности. Способствует развитию творческих способностей, продуктивному сотрудничеству с другими учащимися. Приучает к коллективным действиям, принятию решений, учит руководить и подчиняться, стимулирует практические навыки, развивает воображение.

4. Элементы здоровьесберегающих технологий являются необходимым условием снижения утомляемости и перегрузки учащихся.

5. Проектная технология предлагает практические творческие задания, требующие от учащихся их применение для решения проблемных заданий, знания материала на данный

исторический этап. Овладевая культурой проектирования, школьник приучается творчески мыслить, прогнозировать возможные варианты решения стоящих перед ним задач.

6. Интерактивная: создание специальных заданий, моделирующих реальную жизненную ситуацию, из которой учащимся предлагается найти выход.

7. Информационно-коммуникационные технологии активизируют творческий потенциал учащихся; способствует развитию логики, внимания, речи, повышению качества знаний; формированию умения пользоваться информацией, выбирать из нее необходимое для принятия решения, работать со всеми видами информации, программным обеспечением, специальными программами и т.д.

Организация и мониторинг урока

По санитарным нормам, учащиеся должны работать за компьютером в течение 25 минут в течение урока, поэтому рекомендуется в первой половине урока изучать теорию, выполнять записи, эскизы и чертежи в тетради, работая за партами.

После объяснения нового материала, учащиеся делают гимнастику и садятся к компьютерам. Работают за компьютерами под фронтальным наблюдением учителя. Передают выполненную на компьютере работу по сетевой связи на учительский терминал, а тетрадь на проверку.

№ п\п	Время урока (мин)	Что делают учащиеся	Что делает учитель	Результат
1. Организационный момент	0 – 1 (1 мин)	Готовятся к уроку, включают компьютеры	Организует порядок и технику безопасности	Учащиеся подготавливаются к работе
2. Объяснение нового материала	2 – 8 (6 мин)	Конспектирование в тетрадях и выполнение эскизов	Объясняет новый материал по теме, экспозиция	Схемы и опорные конспекты
3. Работа на компьютере	8 – 33 (25 мин)	Работа за компьютером по теме урока	Фронтальная работа с учащимися класса	Работа по теме урока на компьютере
4. Физкультминутка	33 -34 (1 мин)	Гимнастика для рук (выполняют)	Гимнастика для рук (показывает)	Разрядка
5. Закрепление материала урока	34 – 39 (5 мин)	Отвечают на вопросы, ведут дискуссию	Задаёт вопросы. Даёт объяснения	Дискуссия Закрепление по теме
6. Рефлексия	39 – 40 (2 мин)	Сдача работы Выключение компьютера	Выставление оценок	Делятся впечатлениями

Учитель работает с классом во фронтальном режиме. Дети охотно объясняют друг другу элементы, вызывающие затруднения – этому препятствовать не следует. Можно выделить для себя таких «помощников».

Учёт индивидуальных качеств учащихся. Для медлительных и отстающих учащихся предусматривается ограничение объёма задания; для идущих впереди разработаны варианты заданий повышенной сложности. В качестве поощрения даётся возможность выполнять оригинальные творческие объекты по заданной теме.

Методические формы проведения занятий. Разнообразие приёмов работы на компьютере и большие возможности в вариативном выполнении одного и того же действия в Python, требуют на первых занятиях жёсткого соблюдения алгоритмов.

Коллективные формы работы: предусмотрена возможность выполнять творческие работы по 2 человека по желанию учащихся.

Контрольные занятия рекомендуется проводить в соревновательном режиме.

Темы проектных работ учащиеся обычно находят сами, поэтому рекомендуется сначала активизировать детей, выслушать их тематику. Темы проектирования могут быть выбраны из круга интересов детей и обеспечения учебного процесса:

1. Разработка игры «Шахматы» для двух игроков с консольным интерфейсом
2. Разработка игры «Змейка» с графическим интерфейсом (Tkinter)
3. Разработка игры «Футбол» с графическим интерфейсом
4. Создание приложения для автоматической генерации писем и документов по шаблону MS Word
5. Создание консольного графического редактора (применение фильтров, изменение размеров, повороты изображения и пр.)
6. Разработка приложения «Текстовый редактор»
7. Разработка приложения «Графический редактор»
8. Разработка игры «Гексагон» с графическим интерфейсом
9. Создание чат бота
10. Разработка игры для трех игроков с возможностью добавления ботов
11. Создание WEB-проекта, предоставляющего для команд пользователей совместное рабочее пространство

Участие в Олимпиадах и конкурсах по программированию – хороший стимул для расширения компетенций учащихся в области инженерных решений.

Обучение по этой программе способствует расширению инженерного кругозора учащихся, приучает к самостоятельному принятию решений и обоснованному выбору профессии и специальности.

Предметом диагностики и контроля являются внешние образовательные продукты учеников, а также их внутренние личностные качества (освоенные способы деятельности, знания, умения), которые относятся к целям и задачам курса.

Основой для оценивания деятельности учеников являются результаты анализа его продукции и деятельности по ее созданию. Оценка имеет различные способы выражения — устные суждения педагога, письменные качественные характеристики.

Оценке подлежит в первую очередь уровень достижения учеником минимально необходимых результатов, обозначенных в целях и задачах курса. Оцениванию подлежат также те направления и результаты деятельности учеников, которые определены в рабочей программе учителя и в индивидуальных образовательных программах учеников.

Ученик выступает полноправным субъектом оценивания. Одна из задач педагога — обучение детей навыкам самооценки. С этой целью учитель выделяет и поясняет критерии оценки, учит детей формулировать эти критерии в зависимости от поставленных целей и особенностей образовательного продукта.

Проверка достигаемых учениками образовательных результатов производится в следующих формах:

- ✓ текущий рефлексивный самоанализ, контроль и самооценка учащимися выполняемых заданий - оценка промежуточных достижений используется как инструмент положительной мотивации, для своевременной коррекции деятельности учащихся и учителя; осуществляется по результатам выполнения учащимися практических заданий на каждом занятии;
- ✓ взаимооценка учащимися работ друг друга или работ, выполненных в группах;
- ✓ текущая диагностика и оценка учителем деятельности школьников;
- ✓ итоговый контроль проводится в конце всего курса в форме публичной защиты творческих работ (индивидуальных или групповых);

На основе творческих работ проводятся конкурсы и выставки, формируются «портфолио» учащихся. Данный тип контроля предполагает комплексную проверку

образовательных результатов по всем заявленным целям и направлениям курса. Формой итоговой оценки каждого ученика выступает образовательная характеристика, в которой указывается уровень освоения им каждой из целей курса и каждого из направлений индивидуальной программы ученика по курсу.

Итоговая оценка результатов изучения курса осуществляется по результатам защиты специальной зачетной работы ученика.

УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Успешная реализация предлагаемой программы ориентирована на существующую информационно-образовательную среду школы. Информационно-образовательная среда образовательной организации включает комплекс информационно-образовательных ресурсов, в том числе цифровые образовательные ресурсы, совокупность технологических средств информационных и коммуникационных технологий: компьютеры и иное ИКТ-оборудование, коммуникационные каналы.

Реализация программы основывается на нескольких идеях организации учебно-воспитательного процесса:

- Идея гуманистического подхода предусматривает отношение педагога к обучающемуся как к младшему товарищу, который будет его сменой.
- Идея индивидуального подхода вытекает из учета личностных особенностей, в том числе, в области выбора учащимися задания или характера работы в объединении.
- Идея творческого саморазвития реализуется через побуждение всех детей к самостоятельным исследованиям, самовоспитанию и самосовершенствованию.
- Идея практической направленности осуществляется через сочетание теоретической и практической работы, участия в конкурсах и олимпиадах
- Идея коллективизма опирается на совместную работу групп обучающихся по решению графических задач, коллективное обсуждение теоретических вопросов.

Реализация программы осуществляется на основе следующих принципов:

- Принцип научности, направленный на получение достоверной информации;
- Принцип систематичности и последовательности, требующий логической последовательности в изложении материала;
- Принцип доступности, заключается в необходимой простоте изложения материала;
- Принцип преодоления трудностей предусматривает, что обучающее задание не должно быть слишком простым;
- Принцип сознательности и активности основан на свободном выборе обучающегося направления своей работы.

Для реализации программы выбран язык Python – это один из самых популярных, быстро развивающихся языков в современном мире, при этом он имеет достаточно простой синтаксис и подходит для начального обучения. Читать код на этом языке программирования легко, т.к. в нем используется минимум вспомогательных элементов, а правила языка заставляют программистов делать отступы. В Python есть инструменты, заметно ускоряющие обучение и позволяющие создавать несложные анимации для своих игр, поэтому большинство тем изучается на примерах написания игр. С другой стороны, Python – это универсальный язык программирования. Это язык так называемого сверх высокого уровня – он поддерживает несколько парадигм программирования: структурное, объектно-ориентированное, функциональное и аспектно-ориентированное. Язык Python с легкостью можно применять в любых целях – как для написания небольших скриптов, так и для создания полноценных приложений, от вебпрограммирования до разработки игр. Для языка Python создано множество доступных сред разработки, сервисов и фреймворков. В стандартных библиотеках Python есть средства для работы с электронной почтой, протоколами Интернета, FTP, HTTP, базами данных и другие инструменты. Хорошая модульность языка дает возможность составлять программу из готовых элементов,

выполняющих различные действия. В некоторых случаях для написания программы достаточно лишь найти подходящие модули и правильно их скомбинировать.

Организационно-педагогические условия реализации программы.

Образовательный процесс осуществляется на основе учебного плана, рабочей программы и регламентируется расписанием занятий. В качестве нормативно-правовых оснований проектирования данной программы выступает Федеральный закон Российской Федерации от 29.12.2012 г. №273-ФЗ «Об образовании в Российской Федерации», приказ Министерства образования Российской Федерации от 29.08.2013 г. № 1008 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам», Устав МОУ «Ключевская средняя общеобразовательная школа», правила внутреннего распорядка обучающихся МОУ «Ключевская средняя общеобразовательная школа», локальные акты МОУ «Ключевская средняя общеобразовательная школа». Указанные нормативные основания позволяют образовательному учреждению разрабатывать образовательные программы с учетом интересов и возможностей обучающихся.

Научно-методическое обеспечение реализации программы направлено на обеспечение широкого, постоянного и устойчивого доступа для всех участников образовательного процесса к любой информации, связанной с реализацией общеразвивающей программы, планируемыми результатами, организацией образовательного процесса и условиями его осуществления.

Социально-психологические условия реализации образовательной программы обеспечивают:

- учет специфики возрастного психофизического развития обучающихся;
- вариативность направлений сопровождения участников образовательного процесса (сохранение и укрепление психологического здоровья обучающихся; формирование ценности здоровья и безопасного образа жизни; дифференциация и индивидуализация обучения; мониторинг возможностей и способностей обучающихся, выявление и поддержка одаренных детей, детей с ограниченными возможностями здоровья);
- формирование коммуникативных навыков в разновозрастной среде и среде сверстников.

Разделы программы содержат методические указания по изучению учебного материала раздела, лекционный материал, предлагаемый к изучению, практикум, включающий упражнения для выполнения на компьютере, библиотеку учебной литературы по изучаемой теме, тест промежуточного контроля, содержащий контрольные вопросы по основным терминам и понятиям изучаемого раздела или контрольное задание.

В идеальной модели у учащихся будет воспитана потребность в творческой деятельности в целом и к техническому творчеству в частности, а также сформирована зона личных научных интересов.

Материально-технические условия реализации программы:

- помещение, оборудованное для теоретических и практических занятий «Точка Роста».
- мультимедийная аппаратура для демонстрации видеоматериалов

Аппаратное и техническое обеспечение:

Компьютерный класс в «Точке Роста».

– Рабочее место обучающегося: ноутбук: производительность процессора (по тесту PassMark — CPU BenchMark <http://www.cpubenchmark.net/>): не менее 2000 единиц; объём оперативной памяти: не менее 4 Гб; объём накопителя SSD/eMMC: не менее 128 Гб (или соответствующий по характеристикам персональный компьютер с монитором, клавиатурой и колонками); мышь.

– Рабочее место наставника: ноутбук: процессор Intel Core i5-4590/AMD FX 8350 — аналогичная или более новая модель, графический процессор NVIDIA GeForce GTX 970, AMD Radeon R9 290 — аналогичная или более новая модель, объём оперативной памяти: не менее 4 Гб, видеовыход HDMI 1.4, DisplayPort 1.2 или более новая модель (или

соответствующий по характеристикам персональный компьютер с монитором, клавиатурой и колонками);

- флипчарт с комплектом листов/маркерная доска, соответствующий набор письменных принадлежностей — 1 шт.;

- единая сеть Wi-Fi.

Программное обеспечение:

- Операционная система Windows или Linux;;

- Интерпретатор Python версии 3.7 и выше;

- IDE JetBrains PyCharm;

- Foxit Reader или другой просмотрщик PDF файлов;

- Пакет офисных программ;

- Любой браузер для интернет серфинга.

СПИСОК ЛИТЕРАТУРЫ

для учащегося

1. Задачи по программированию. Под ред. С. М. Окулова, М.: БИНОМ. Лаборатория знаний, 2006.
2. С. М. Окулов. Основы программирования. М.: Бином. Лаборатория знаний, 2012.
3. Майк МакГрат «Программирование на Python для начинающих» Эксмо, 2015.
4. Федоров Д. Ю. Основы программирования на примере языка Python. //Учебное пособие. – Санкт-Петербург: 2016.
5. Сэнд У., Сэнд К. «Hello World! Занимательное программирование на языке Python» - М.: – 2016.
6. Сайт / интерактивный сборник задач для практики программирования [Электронный ресурс] – Режим доступа: <http://pythontutor.ru/>, свободный.
7. Сайт / Адаптивный тренажер Python [Электронный ресурс] – Режим доступа: <https://stepik.org/course/431>, свободный.
8. Сайт / pythonworld.ru — «Python 3 для начинающих»

для педагога

1. Бизли, Дэвид М. Python. Подробный справочник. – М.–СПб.: Символ-Плюс, 2010.
2. Лутц, Марк Python. Справочник. – М.: Вильямс, 2015.
3. Сэнд У., Сэнд К. «Hello World! Занимательное программирование на языке Python» - М.: – 2016.
4. Долинский М.С. Решение сложных и олимпиадных задач по программированию - Учебное пособие - М.: – 2006.
5. Россум Г., Дж. Дрейк Ф.Л., Откидач Д.С. Язык программирования Python. 2010.
6. Щерба А.В. Изучение языка программирования Python на основе задач УМК авторов И.А. Калинин и Н.Н. Самылкина. //Учебное пособие. –М.: МПГУ, 2015.
7. Официальный сайт программы [Электронный ресурс] – Режим доступа: <https://docs.python.org/>, свободный.
8. Сайт, среда разработки для языка Python. [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.
9. <http://informatics.mccme.ru/course/view.php?id=156>
10. https://inf5.ru/podgotovka_k_olympiad/olym_zadachi_s_resheniyami.htm
11. <http://anngorg.ru/olimp/materials>
12. <http://kpolyakov.spb.ru/school/probook/python.htm>
13. <http://dist-olimpiada.krasnogorka.edusite.ru/p4aa1.html>
14. Сайт / справочные материалы [Электронный ресурс] – Режим доступа: <https://metanit.com/python/>, свободный.
15. Сайт / среда разработки для языка Python [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.
16. Сайт проекта Open Book Project. Практические примеры на Python Криса Мейерса [Электронный ресурс] – Режим доступа: openbookproject.net, свободный.

Приложение. Оценочный лист

№	ФИ обучающегося	Теоретические знания по основным разделам учебно-тематического плана программы	Практические умения и навыки по основным разделам учебно-тематического плана программы	Творческие Креативность в выполнении практических заданий
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
...				

Приложение. Протокол результатов аттестации обучающихся (групповой)

Уровень обучения										
Учебный год										
	Вид аттестации	входной контроль	текущая аттестация	промежуточная аттестация	входной контроль	текущая аттестация	промежуточная аттестация	входной контроль	текущая аттестация	промежуточная аттестация
№	Форма аттестации Ф.И. обучающегося									
1.										
2.										
3.										
4.										
5.										
6.										
7.										
8.										
9.										
10.										
11.										
12.										
ИТОГИ АТТЕСТАЦИИ										
Высокий уровень (чел.)										
Средний уровень (чел.)										
Низкий уровень (чел.)										
ВСЕГО чел.										
ПОДПИСЬ ПЕДАГОГА:										
Подписи членов аттестационной комиссии										

**Приложение. Лист корректировки изменений в дополнительной общеобразовательной
общеразвивающей программе**

Тема раздела	Кол-во часов по плану	Проведено фактически	Дата и причина коррекции (б/л, праздники, КПК и т.д.)

Входящий контроль

Низкий уровень: 0-70% выполненных заданий;

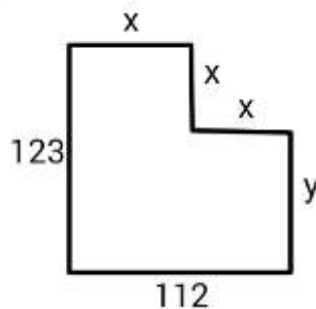
Средний уровень: 70-85% правильно выполненных заданий;

Высокий уровень: 85-100% правильно выполненных заданий.

Для учащихся – 14-17 лет

Фамилия, Имя	
---------------------	--

1. Из прямоугольника вырезали квадрат со стороной x , получилась фигура как на рисунке. Чему равна сумма цифр y ?



2. В алфавите племени мумба-юмба 32 буквы. Любое слово в языке этого племени состоит из пяти букв и должно одинаково читаться справа налево и слева направо, при этом первые две буквы слова обязательно различаются, а третья совпадает с пятой.

Каково максимальное количество слов в этом языке?

3. Катя наклеила на рулет тонкие поперечные кольца трёх разных цветов. Если разрезать по серым кольцам, получится 25 кусков рулета, если по малиновым — 47 кусков, а если по зеленым — 31 кусок.

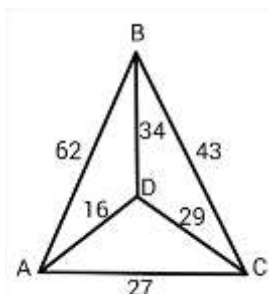
Сколько кусков рулета получится, если разрезать по кольцам всех трёх цветов?

Примечания

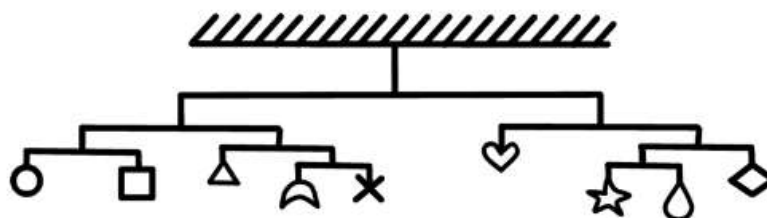
Учтите, что наклеить кольцо одного цвета на кольцо другого нельзя.

4. На рисунке показано расположение городов А, В, С и D и расстояния между ними. Турист выходит из города В и собирается посетить остальные города, побывав в каждом по разу.

Какова наименьшая возможная длина маршрута, если он хочет закончить свой путь в том же городе?



5. Фигурки, общей массой 432 грамма, при помощи невесомых нитей и планок собрали в конструкцию, изображённую на рисунке. Оказалось, что все её части находятся в равновесии. Сколько весит ромб?



6. Люди переезжают в города, за год численность людей удваивается. Если люди заселят весь город за 12 лет, то сколько лет понадобится, чтобы занять лишь половину города?

7. Сравните пары слов. Сколько среди них полностью идентичных?

O/Sanmarco

O/Samnarco

Ф.Wagonerrte

Ф.Wagonertre

A.S.Schmetterling

A.S.Schnetterling

N.V.Murfreesboroque

N.V.Munfreesboroque

P.S.Splendoursec

P.S.Sqlendoursec

Семь человек выясняли, какой сегодня день недели.

Первый сказал: «Послезавтра – воскресенье».

Второй: «Вчера был понедельник».

Третий: «Завтра будет суббота».

Четвертый: «Завтра будет среда».

Пятый: «Вчера был четверг».

Шестой: «Позавчера было воскресенье».

Седьмой: «Позавчера была среда».

Какой сегодня день недели, если трое ошибаются?

8. Вам предложены несколько высказываний и следствие из них (выделено жирным).

Согласны ли Вы с этим следствием?

1. Все клёны — растения.

2. Некоторые растения быстро желтеют.

Значит, некоторые клёны быстро желтеют.

Да

Нет

9. Гусеница прогрызает яблоко диаметром 6 сантиметров насквозь за 16 секунд, вылезая снаружи полностью.

Известно, что середину яблока она начинает грызть уже через 6 секунд после начала пути.

Какова длина гусеницы в сантиметрах?

Для учащихся 12-13 лет

Фамилия, Имя.	
---------------	--

1. Назовите два числа, у которых количество цифр равно количеству букв, составляющих название каждого из этих чисел.

--	--

2. Собака была привязана к десятиметровой веревке, а прошла двести метров. Как ей это удалось?

--

3. Есть дорога, по которой может проехать только одна машина. По дороге едут две машины: одна с горы, другая под гору. Как им разъехаться?

--

4. Как с помощью только одной палочки образовать на столе треугольник?

--

5. В каком месяце болтливая девочка говорит меньше всего?

--

6. Что становится на треть больше, если его поставить вверх ногами?

--

7. Представьте себе, что вы кондуктор. Поезд везет сто вагонов, в каждом вагоне 10 купе, в каждом купе 4 пассажира. Сколько лет кондуктору?

--

8. Юра разрезал огромную пиццу на 10 кусков. Затем он взял один из кусков и разрезал его еще на 10. После этого из имеющихся кусков он выбрал два и разрезал каждый из них на 10.

Вопрос: Сколько кусков пиццы получилось у Юры.

--

9. В компьютерной игре нужно победить монстра. Изначально у Юры было только 9 выстрелов. Но за каждое попадание он получал дополнительно еще 3 выстрела.

Вопрос: Сколько раз Юра попал в цель, если всего он выстрелил 30 раз, израсходовав все выстрелы?

--

Итоговый контроль
Диагностическая карта

№/ п	Имя	Входящая Диагностика	Промежуточ- ная диагностика	Итоговая диагностика
1				
2				
3+				
	Итого в %	Н С В		Н С В

Результативность отслеживается с помощью карты наблюдений, анализа участия детей в совместной продуктивной деятельности, разработки и защите творческих проектов. Усвоение программы возможно по 3-м уровням: низкий (Н), средний (С), высокий (В).

Низкий уровень

Учащиеся должны знать / понимать:

- технику безопасного поведения во время занятий;
- правила поведения в общественных местах,
- понятие программы Python; общую структуру программы;
- основные типы данных;
- оператор присваивания;
- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- основные циклы с условием;
- основные правила записи циклов условием;
- формат записи цикла с параметром;
- понятие функции;
- основные принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- способ передачи параметров.
- назначение строкового типа данных;
- операторы для работы со строками;
- операции со строками;
- способ описания списка;
- способ описания кортежа;
- способ описания словаря;
- основные операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.
- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ.

Учащиеся должны уметь:

- понимать учебную задачу, сохранять ее содержание в процессе ее выполнения под руководством педагога;
- работать в паре, малой группе;
- выполнить установку программы под руководством педагога;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе под руководством педагога;

- решать задачи на элементарные действия с числами;
- использовать условный оператор;
- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием под руководством педагога;
- определять целесообразность применения и использования цикла с параметром для решения поставленной задачи по наводящим вопросам педагога;
- создавать и использовать основные функции;
- описывать и соединять строки;
- находить подстроку в строке с помощью педагога;
- находить количество слов в строке;
- вводить и выводить элементы списка под руководством педагога;
- приводить примеры использования вложенных списков (матриц) по наводящим вопросам педагога;
- описывать множества под руководством педагога;
- определять принадлежность элемента множеству по наводящим вопросам;
- определять вид ошибок и находить ошибки в программе под руководством педагога;
- составлять элементарные алгоритмы для решения задач;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python под руководством педагога.

Средний уровень

Учащиеся должны знать / понимать:

- основные положения техники безопасности на занятиях, правила поведения в общественных местах, правила дорожной безопасности;
- основные приемы взаимодействия в группе сверстников;
- понятие программы;
- структуру программы на Python;
- режимы работы с Python.
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;
- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- циклы с условием и их виды;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.
- понятие функции;
- основные способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.
- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками;
- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;

- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.
- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ;
- основные шаги работы над проектом, его презентации.

Учащиеся должны уметь:

- уважительно относиться к преподавателям и сверстникам;
- применять некоторые приемы логического (абстрактное) мышления;
- концентрировать внимание на одном или двух объектах;
- понимать причины успеха/неуспеха с помощью анализа педагога;
- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе;
- решать задачи на элементарные действия с числами;
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов;
- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использования цикла с параметром для решения поставленной задачи;
- создавать и использовать функции;
- использовать механизм параметров для передачи значений;
- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке;
- описывать списки;
- вводить элементы списка;
- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.
- определять вид ошибок и находить ошибки в программе.
- составлять алгоритмы для решения задач;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- отлаживать и тестировать программы, написанные на языке Python;
- понимать и сохранять цели и задачи учебной деятельности в ходе занятия;
- планировать свою деятельность с помощью взрослого;
- сотрудничать с взрослыми и сверстниками в процессе выполнения учебной задачи;

- понимать причины успеха/неуспеха учебной деятельности под руководством взрослого;
- делать выводы в ходе полученных заданий; выстраивать логические цепи рассуждений под руководством педагога;
- выражать творческие идеи, разработать творческий проект на основе образца;
- конструктивно взаимодействовать в составе группы в ходе работы над проектом.

Высокий уровень

Учащиеся должны знать / понимать:

- основные положения техники безопасности на занятиях, правила поведения в общественных местах, правила дорожной безопасности, правила поведения во время чрезвычайных происшествий;
- приемы конструктивного взаимодействия в группе сверстников;
- понятие программы;
- структуру программы на Python;
- режимы работы с Python.
- общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;
- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- циклы с условием и их виды;
- правила записи циклов условием;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.
- понятие функции;
- способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.
- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками;
- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.
- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ;
- правила и этапы работы над проектом;

- приемы успешной презентации проекта.
- Учащиеся должны уметь:*
- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе;
- решать задачи на элементарные действия с числами;
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов;
- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использования цикла с параметром для решения поставленной задачи;
- создавать и использовать функции;
- использовать механизм параметров для передачи значений;
- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке;
- описывать списки;
- вводить элементы списка;
- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.
- определять вид ошибок и находить ошибки в программе.
- составлять алгоритмы для решения задач;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- отлаживать и тестировать программы, написанные на языке Python;
- понимать учебную задачу, анализировать достижение результата;
- делать выводы в ходе полученных заданий; самостоятельно выстраивать логические цепи рассуждений;
- понимать причины успеха/неуспеха, конструктивно принимать причины неуспеха, исправлять ошибки с минимальным участием педагога;
- планировать свою деятельность, находить оригинальные способы выполнения поставленной творческой задачи;
- создать проектировочную команду и организовать ее деятельность;
- разработать творческий проект по предложенной или самостоятельно выбранной теме в составе творческой группы;
- самостоятельно подготовить презентацию и защитить проект.

Вопросы для промежуточного контроля по усвоению материала

Составьте выражение для вычисления в интерпретаторе Python 3 и вставьте в поле ответа результат вычисления:

11111·1111111

— произведение чисел 11111 (5 единиц) и 1111111 (7 единиц)

Запишите число 1.2345e3 в виде десятичной дроби.

Составьте и запишите выражение для вычисления:

2014.0^14(Возвестив 14 степень)

Обратите внимание на запись числа: это вещественное число.

Приведите к целому типу число 2.99

Расставьте скобки в выражении

a and b or not a and not b

в соответствии с порядком вычисления выражения (приоритетом операций). Всего потребуется 5 пар скобок (внешние скобки входят в их число).

Найдите результат выражения для заданных значений *a* и *b*

Учитывайте регистр символов при ответе.

a = True

b = False

a and b or not a and not b

Отметьте выражения, значения которых равны True:

- "239" < "30" and 239 < 30
- "239" < "30" and 239 > 30
- "239" > "30" and 239 < 30
- "239" > "30" and 239 > 30

Укажите результат выражения:

"123" + "42"

Какое значение будет у переменной *i* после выполнения фрагмента программы?

```
i = 0
while i <= 10:
    i = i + 1
    if i > 7:
        i = i + 2
```

Сколько итераций цикла будет выполнено в этом фрагменте программы?

```
i = 0
while i <= 10:
    i = i + 1
    if i > 7:
        i = i + 2
```

Сколько всего знаков * будет выведено после исполнения фрагмента программы:

```
i = 0
while i < 5:
    print('*')
    if i % 2 == 0:
```

```
print('**')  
if i > 2:  
    print('***')  
i = i + 1
```

Определите, какое значение будет иметь переменная *i* после выполнения следующего фрагмента программы:

```
i = 0  
s = 0  
while i < 10:  
    i = i + 1  
    s = s + i  
    if s > 15:  
        break  
    i = i + 1
```

Определите, какое значение будет иметь переменная *i* после выполнения следующего фрагмента программы:

```
i = 0  
s = 0  
while i < 10:  
    i = i + 1  
    s = s + i  
    if s > 15:  
        continue  
    i = i + 1
```


Задачи для промежуточного контроля по усвоению материала

<p>Напишите простой калькулятор, который считывает с пользовательского ввода три строки: первое число, второе число и операцию, после чего применяет операцию к введённым числам ("первое число" "операция" "второе число") и выводит результат на экран.</p> <p>Поддерживаемые операции: +, -, /, *, mod, pow, div, где mod — это взятие остатка от деления, pow — возведение в степень, div — целочисленное деление.</p> <p>Если выполняется деление и второе число равно 0, необходимо выводить строку "Деление на 0!".</p> <p>Обратите внимание, что на вход программе приходят вещественные числа.</p>
<p>Жители страны Малевии часто экспериментируют с планировкой комнат. Комнаты бывают треугольные, прямоугольные и круглые. Чтобы быстро вычислять жилплощадь, требуется написать программу, на вход которой подаётся тип фигуры комнаты и соответствующие параметры, которая бы выводила площадь получившейся комнаты.</p> <p>Для числа π в стране Малевии используют значение 3.14.</p> <p>Формат ввода, который используют Малевийцы:</p> <p>Треугольник a b c <i>где a, b и c — длины сторон треугольника</i></p> <p>прямоугольник a b <i>где a и b — длины сторон прямоугольника</i></p> <p>круг r <i>где r — радиус окружности</i></p>
<p>Напишите программу, которая получает на вход три целых числа, по одному числу в строке, и выводит на консоль в три строки сначала максимальное, потом минимальное, после чего оставшееся число.</p> <p>На ввод могут подаваться и повторяющиеся числа.</p>
<p>Паша очень любит кататься на общественном транспорте, а получая билет, сразу проверяет, счастливый ли ему попался. Билет считается счастливым, если сумма первых трех цифр совпадает с суммой последних трех цифр номера билета.</p> <p>Однако Паша очень плохо считает в уме, поэтому попросил вас написать программу, которая проверит равенство сумм и выведет "Счастливый", если суммы совпадают, и "Обычный", если суммы различны.</p> <p>На вход программе подаётся строка из шести цифр. (Пример:123321)</p> <p>Выводить нужно только слово "Счастливый" или "Обычный", с большой буквы.</p>
<p>Напишите программу, которая считывает с консоли числа (по одному в строке) до тех пор, пока сумма введённых чисел не будет равна 0 и сразу после этого выводит сумму квадратов всех считанных чисел.</p> <p>Гарантируется, что в какой-то момент сумма введённых чисел окажется равной 0, после этого считывание продолжать не нужно.</p> <p>В примере мы считываем числа 1, -3, 5, -6, -10, 13; в этот момент замечаем, что сумма этих чисел равна нулю и выводим сумму их квадратов, не обращая внимания на то, что остались ещё не прочитанные значения.</p>
<p>Напишите программу, которая выводит часть последовательности 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 ... (число повторяется столько раз, чему равно). На вход программе передаётся</p>

неотрицательное целое число n — столько элементов последовательности должна отобразить программа. На выходе ожидается последовательность чисел, записанных через пробел в одну строку.

Например, если $n = 7$, то программа должна вывести 1 2 2 3 3 3 4.

Выведите таблицу размером $n \times n$, заполненную числами от **1 до n^2** по спирали, выходящей из левого верхнего угла и закрученной по часовой стрелке, как показано в примере (**здесь $n=5$**)