

РОССИЙСКАЯ ФЕДЕРАЦИЯ
АДМИНИСТРАЦИЯ ГОРОДСКОГО ОКРУГА «ГОРОД КАЛИНИНГРАД»
КОМИТЕТ ПО ОБРАЗОВАНИЮ
муниципальное автономное учреждение дополнительного образования
города Калининграда Станция юных техников

Принята на заседании
педагогического совета МАУДО СЮТ
протокол № ____ от «____» _____ 2026 г.

УТВЕРЖДАЮ
Директор МАУДО СЮТ
_____/Стецюк О.Е./
«____» _____ 2026 г.

**Дополнительная общеобразовательная общеразвивающая программа
технической направленности «Поколение Python»**

Возраст обучающихся: 12-17 лет

Срок реализации: 2 года

Автор программы:
Трущелев Александр Николаевич,
педагог дополнительного образования
г. Калининград

Калининград, 2026

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Описание предмета, дисциплины которому посвящена программа

Программа “Поколение Python” служит для обучения детей среднего и старшего школьного возраста основам программирования. Для этого используются возможности языка высокого уровня Python.

Python последние восемь лет занимает лидирующие места в индексе ТЮВЕ (рейтинг популярности языков программирования). Востребованность языка объясняется широтой сфер применения: научные исследования, веб-разработка, Big Data, машинное обучение, системное администрирование, искусственный интеллект, игры и многое другое. Специалисты, владеющие Python, востребованы у работодателей, так как язык используют стартапы, крупные компании, транснациональные корпорации, госслужбы. При этом зарплата специалистов, имеющих опыт от двух лет, начинается от 100 000 рублей.

Благодаря ясному синтаксису изучение языка не составляет большой проблемы. Написанные на нем программы получают структурированными по форме, и в них легко проследить логику работы.

Python свободно доступен для многих платформ, а написанные на нем программы обычно переносимы между платформами без изменений. Это обстоятельство позволяет применять для изучения языка любую имеющуюся аппаратную платформу.

Раскрытие ведущих идей, на которых базируется программа

Программа базируется на нескольких идеях, которые определяют ее структуру и содержание. Вот некоторые из них:

1. Практическое применение: Программа стремится предоставить учащимся возможность немедленно применить изучаемые концепции и навыки на практике. Вместо того, чтобы только теоретически объяснять синтаксис и конструкции языка Python, акцент делается на создании практических проектов и упражнений, которые помогут учащимся увидеть, как программирование может быть использовано для решения реальных задач.

2. Логическое мышление и решение проблем: Программа ставит перед собой задачу развития логического мышления учащихся и способности анализировать и решать проблемы с использованием программирования. Учащиеся будут поощряться к разбору сложных задач на более простые компоненты, исследованию различных подходов и применению алгоритмического мышления для нахождения эффективных решений.

3. Актуальность и практическая применимость: Программа разработана с учетом актуальности языка программирования Python и

его практической применимости в современном мире. Python является одним из самых популярных языков программирования в индустрии и используется для разработки веб-приложений, научных исследований, анализа данных, искусственного интеллекта и многого другого. Программа помогает учащимся освоить навыки, которые востребованы на рынке труда.

4. Дифференцированный подход: Учитывая различные возрастные группы учащихся, программа учитывает разные уровни подготовки и интересы. Материалы и проекты могут быть адаптированы в зависимости от уровня учащихся, чтобы обеспечить оптимальное обучение и удовлетворить индивидуальные потребности.

5. Интерактивное обучение: Программа включает использование различных педагогических технологий, методов и приемов, чтобы сделать процесс обучения интерактивным и увлекательным. Это может включать в себя использование визуальных презентаций, интерактивных упражнений, головоломок, игр и коллективных проектов. Такой подход позволяет учащимся активно взаимодействовать с материалами и участвовать в процессе обучения.

Эти идеи служат основой логики изложения материалов программы, выбора тем и использования педагогических методов и приемов. Они помогают обеспечить полезность и эффективность программы, а также создать современную, практико-ориентированную, высокотехнологичную образовательную среду, позволяющую эффективно реализовывать обучающую, проектную и экспериментальную исследовательскую деятельность учащихся в разновозрастных группах, получать новые образовательные результаты и инновационные продукты.

Описание ключевых понятий, которыми оперирует автор программы

В программе имеется ряд ключевых понятий, связанных с программированием на языке Python и компьютерными науками. Вот описание некоторых из них:

Python: Python это высокоуровневый язык программирования, который отличается простотой, читаемостью и гибкостью. Учащиеся узнают основные конструкции и синтаксис Python, а также научатся использовать его для создания программ и решения задач.

Язык программирования: это формальный язык, предназначенный для записи компьютерных программ.

Программирование: Это процесс создания компьютерных программ, путем написания кода на языке программирования.

Алгоритм: Это точный набор инструкций, описывающих порядок действий некоторого исполнителя для достижения результата, решения некоторой задачи за конечное число шагов.

Инструкция: Это команда или набор команд на языке программирования.

Переменные: Переменная это символическое имя, которое используется для хранения и обращения к данным в программе. Она является именованным контейнером для хранения значения определенного типа данных. Переменные в программировании используются для сохранения информации, которая может изменяться в процессе выполнения программы.

Условные операторы: Позволяют программе принимать решения на основе заданных условий. Они предоставляют средства для выполнения различных действий в зависимости от того, выполняется ли определенное условие или нет.

Циклы: Позволяют выполнять повторяющиеся действия в программе, помогая автоматизировать повторяющиеся задачи и обрабатывать коллекции данных.

Структуры данных: В программировании структуры данных используются для организации и хранения данных (списки, кортежи, словари и др.).

Функции: Функции в программировании представляют собой независимые блоки кода, которые могут быть вызваны для выполнения определенных действий или вычислений. Функции позволяют организовать код программы в логические блоки, которые могут быть многократно использованы и повторно вызываться из разных частей программы.

Это лишь некоторые из ключевых понятий.

Направленность (профиль) программы

Разноуровневая дополнительная общеобразовательная общеразвивающая программа «Поколение Python» имеет **техническую направленность** и практико-ориентированный характер обучения.

Уровень освоения программы

Уровень освоения программы – базовый.

Первый год обучения посвящен основам программирования на языке Python. Он состоит из 8 разделов.

Второй год обучения посвящен также основам программирования на языке Python, но осваиваемые понятия более сложные и основаны на изученном материале первого года обучения. Он состоит из 11 разделов.

Каждый раздел состоит из некоторого количества уроков, которые представляют собой наборы лекций, конспектов и заданий. Обычно один урок посвящен одному понятию. Очень важно после прочтения изучения нового материала выполнить задания.

Все задачи можно решать многократно. За неудачные попытки баллы не снижаются, чтобы ученики не боялись ошибаться!

В конце некоторых разделов есть итоговые работы на время. Они помогут понять, насколько хорошо усвоена тема.

Актуальность программы

Актуальность программы обусловлена тем, что без новых информационных технологий, компьютерных систем уже невозможно представить современное общество, которое ставит перед образованием новые цели и ориентиры по подготовке детей к жизни в условиях быстрых инновационных перемен. Процесс глубоких перемен, происходящих в современном образовании, выдвигает в качестве приоритетной проблему развития творчества, креативного мышления, способствующего формированию разносторонне-развитой личности, отличающейся неповторимостью, оригинальностью, умеющей решать сложные задачи. Компьютерные системы, индивидуальные, промышленные, военные роботы, «умная» техника, простые и сложные программы, системы управления целыми предприятиями, космические аппараты все это и многое другое требует управляющих программ. Без знания языков программирования невозможно представить успешного человека будущего.

Язык Python довольно прост и удобен для освоения, но в тоже время очень мощный, отлично спроектирован, логичен и универсален, поэтому подходит для решения разнообразных задач и многих платформ. Разработка на нем идёт гораздо быстрее, т.к. кода здесь меньше, чем на других языках. Это самый популярный язык программирования, в настоящее время.

Он используется для веб-разработки (Pyramid, Pylons, TurboGears, Flask, CherryPy, Django), и на мобильных устройствах, в приложениях и решениях, связанных с машинным обучением (нейросети и искусственный интеллект), в играх (Battlefield 2, World of Tanks, Civilization IV, EVE Online). Язык применяется во встроенных системах станков с ЧПУ, для управления банкоматами, в средствах автоматического регулирования и в телекоммуникационном оборудовании. Системным администраторам Python нужен для автоматизации задач. Он простой, мощный и поддерживает специальные пакеты, которые повышают его эффективность. И, самое главное, он по умолчанию установлен на все серверы с ОС Linux. В Python есть несколько библиотек, которые пригодятся для проведения научных исследований и вычислений (SciPy, NumPy, Matplotlib), благодаря библиотекам и простоте освоения языка многие учёные выбирают Python — особенно он популярен у математиков и физиков. Python — один из самых используемых в Data Science языков. На нём пишут алгоритмы программ с машинным обучением и аналитические приложения. С помощью него обслуживают хранилища данных и облачные сервисы. Очень многие ведущие компании используют Python в своей деятельности — Сбербанк, Mail.ru, Яндекс, Google, Pixar, Industrial Light & Magic, Intel, Cisco, HP, Seagate, Qualcomm, IBM, JPMorgan Chase, UBS, Getco, Citadel, NASA, Агентство национальной безопасности США, Instagram, Positive Technologies, Houdini, Facebook, Yahoo, Red Hat, Dropbox, Pinterest и многие другие. Современные технологии просто невозможно представить без программ.

По результатам исследования компании Stack Overflow (<https://survey.stackoverflow.co/2024/>) язык Python лидер среди самых используемых языков программирования в 2023 году и один из самых востребованных на рынке труда. Также в индексе языков программирования (TIOBE <https://www.tiobe.com/tiobe-index/>) Python уже несколько лет занимает 1-е место и с каждым годом увеличивает разрыв. Изучение Python в школе откроет ученикам возможности дальнейшего развития в области IT и поможет профориентации в старших классах.

Педагогическая целесообразность

Педагогическая целесообразность программы заключается в том, что изучение и применение знаний по программированию на Python в учебном процессе формирует инженерный подход к решению задач, дает возможность развития творческого мышления у детей, привлекает школьников к исследованиям в межпредметных областях, особенно в информатике, математике, теории вероятностей, физике, радиоэлектронике, робототехнике. При реализации программы применяется как групповая форма работы, так и индивидуальная. Для того чтобы занятия были максимально интересными применяются элементы геймификации, в виде онлайн викторин при проверке знаний. Программа «Поколение Python» дает возможность продолжить обучение в области программирования для учащихся старших классов, обладает возможностями расширения возможностей для проектной и исследовательской деятельности учащихся в различных предметных областях, а также позволяет развить и применить на практике знания, полученные на занятиях. Эти знания становятся для ребят необходимой теоретической и практической основой их дальнейшего участия в техническом творчестве, выборе будущей профессии.

Практическая значимость

Программа «Поколение Python» - это не только изучение основ программирования на языке Python, но и средство формирования новых общеинтеллектуальных умений и навыков.

Во первых: Программирование развивает вычислительное мышление. А оно учит формулировать проблему, искать решение и анализировать его. Программирование — часть вычислительного мышления и самый эффективный способ его освоить. Например, перед программированием мы анализируем и разбиваем задачу на подзадачи. Программирование конкретизирует вычислительное мышление и может стать инструментом получения знаний.

Во вторых: Программирование развивает конструкционизм. Который объединяет два типа конструирования: дети конструируют в реальном мире и конструируют новые идеи в своей голове. Два типа конструирования создают непрерывную спираль обучения: когда у ребенка

появляются новые идеи, он конструирует новые вещи (программы) в реальном мире. Программирование помогает детям воплотить эти идеи в жизнь.

В третьих: Развивает навыки решения задач.

Программирование и вычислительное мышление помогают детям учиться сетевым взаимоотношениям: общаться в чатах, видеоконференциях, соцсетях. Решая сложные задачи, дети используют четыре основных компонента вычислительного мышления: декомпозицию, выделение паттернов, автоматизацию и абстрагирование.

В четвертых: Стимулирует креативное мышление.

Программирование помогает поддерживать и стимулировать креативность, открывает источники вдохновения, чтобы выражать свои идеи и раскрывать личность.

В пятых: Развивает умение учиться

Обучаясь программированию, дети учатся мыслить по-новому. Если написанная программа не работает или работает не так, как ожидал ребенок, ему приходится разбираться и придумывать решение задачи. Ребенок начинает понимать новые идеи и учиться учиться.

По сравнению с другими видами учебной деятельности программирование — более щадящий и мягкий процесс. Работа с кодом помогает учиться на своих ошибках и не вызывает дискомфорта, если дети их совершают. Например, ребенок может отменить команду, чтобы исправить ошибку в коде, внести правки и попробовать что-то новое. Благодаря программированию дети перестают бояться ошибок.

В шестых: Стимулирует учебную мотивацию

Обычный результат программирования — это проект. Ребенок сразу видит: работает или не работает программа. Если не работает, он старается понять, что нужно узнать для завершения проекта. Ученик мотивирован, ему интересно искать ответы на вопросы, на которые, иной раз, и учитель не знает ответа, но точно может помочь их найти.

Обучать детей программированию нужно не для того, чтобы они в будущем стали программистами. Задача программирования — развить у детей мышление, конструкционизм, навыки решения задач, креативность и умение учиться, которые помогут им уверенно справляться с комплексными, сложными задачами XXI века, не имеющими однозначно правильного решения.

Принципы отбора содержания

Изложение программы «Поколение Python» ведется последовательно, от простого к сложному. Программное обеспечение русифицировано, что позволяет легко и быстро его освоить. Так как программа строится по логическим законам, то возможна организация разнообразной интересной деятельности с четким переходом от одного вида работы к другому, с конкретными указаниями, на что обратить внимание. При этом будет

развиваться произвольное внимание детей. Несмотря на общие возрастные особенности, каждый ребенок индивидуален в своем развитии, поэтому программа предусматривает индивидуальный подход к каждому ребенку.

Творческое, самостоятельное выполнение практических заданий, задания в форме описания поставленной задачи или проблемы, дают возможность учащемуся самостоятельно выбирать пути ее решения. Содержание дополнительного образования в области робототехники не стандартизируется, работа с обучающимся происходит в соответствии с его интересами, его выбором, что позволяет безгранично расширять его образовательный потенциал. При этом реализуются диалоговый характер обучения, приспособление оборудования и инструмента к индивидуальным особенностям ребенка, возможность коррекции педагогом процесса обучения в любой момент, оптимальное сочетание индивидуальной и групповой работы. Данная программа полностью соответствует личностно-ориентированной модели обучения и предоставляет широкие возможности для выявления, учета и развития творческого потенциала каждого ребенка, вкуса, проявления его индивидуальности, инициативы, формирования духовного мира, этики общения, навыка работы в творческом объединении.

Отличительные особенности программы

Программа «Поколение Python» является большим шагом к началу изучения современных информационных технологий и рассчитана на 2 года обучения. Оба года обучения представлены, как циклы, имеющие задачи, учебный план, содержание программы, планируемые результаты.

Курс предполагает смешанный формат обучения. Сочетание групповой работы с учителем в классе, индивидуальной работы в личном кабинете на онлайн-платформе и интерактивных проверочных заданий в онлайн приложениях, позволяет ученикам выработать не только технические навыки программирования, но и навыки социального взаимодействия при работе над проектом, а главное – научиться самостоятельно выстраивать свое профессиональное развитие.

Цель образовательной программы

Цель программы: раскрытие интеллектуального и творческого потенциала детей с использованием возможностей программы.

Задачи образовательной программы

Для достижения поставленной цели решаются следующие задачи:

Обучающие:

- составление алгоритмов для решения поставленных задач;

- изучение возможностей программ для целей программирования на языке Python;
- знакомство учащихся с комплексом базовых технологий, применяемых при программировании;
- приобретение навыков и умений в области конструирования и оптимизации алгоритмов и программ;
- приобретение опыта создания законченных программ.

Развивающие:

- развитие навыков для работы с учебной онлайн платформой.
- развитие навыков работать с учебной литературой и справочными файлами, ориентироваться в информационном пространстве, анализировать, обобщать, делать выводы;
- формирование ответственного отношения к информации с учетом правовых и этических аспектов ее распространения;
- развитие творческого подхода к решению поставленных задач, умения работать индивидуально и в команде;
- развитие логического, образного и абстрактного мышления;
- развитие интереса к современной технике, конструированию, моделированию, программированию.

Воспитательные:

- создание необходимых условий для личностного развития учащихся, позитивной социализации и профессионального самоопределения;
- удовлетворение индивидуальных потребностей учащихся в интеллектуальном, художественно-эстетическом, нравственном развитии;
- формирование и развитие творческих способностей учащихся, выявление, развитие и поддержку талантливых учащихся;
- формирование культуры здорового и безопасного образа жизни, укрепление здоровья учащихся.

Психолого-педагогические характеристики обучающихся, участвующих в реализации образовательной программы.

Дополнительная общеобразовательная общеразвивающая программа «Поколение Python» предназначена для детей от 12 до 17 лет.

Для успешного обучения по программе требуются средние и высокие когнитивные способности, поэтому набор детей в объединение осуществляется на основании собеседования и входного тестирования.

Особенности организации образовательного процесса

Учащиеся принимаются в группу, без каких-либо первоначальных знаний в области программирования. Количество обучающихся в группе 10-15 человек, обучение смешанное, очное с преподавателем и индивидуальная работа в личном кабинете на онлайн платформе.

Вид деятельности группы – профильная, состав - постоянный.

Задания, содержащиеся в программе, адаптированы к возрасту, курс доступен школьнику средних и высоких способностей.

Формы обучения по образовательной программе

Весь учебный материал, содержащийся в программе, можно разделить на теоретический и практический. Теоретическим вопросам отводится не более 15-20 минут на каждом занятии. Теоретическая часть проходит в виде лекций, разбираются возможные задачи и методы их решения, практическая часть – закрепление пройденного материала, проверка знаний обучающихся, а также выполнение практических заданий.

Преимущество отдается занятиям, как форме организации педагогического процесса, так как они имеют возможности для сочетания фронтальной, групповой и индивидуальной работы; позволяют педагогу систематически и последовательно излагать материал.

Методика проведения занятий направлена на решение основных концептуальных задач обучения. Обучающая программа разбита на разделы и учебные элементы. Преимуществом использования учебных элементов является возможность эффективной реализации принципов:

- индивидуализации работы обучаемых;
- гибкой организации учебного процесса;
- постоянной обратной связи учебного процесса;
- интенсификации учебной деятельности.

На занятиях используется индивидуальный подход к каждому ребенку, особенно при подготовке самостоятельных проектов.

- Вводное занятие – педагог знакомит обучающихся с техникой безопасности, особенностями организации обучения и предлагаемой программой работы на учебный год. На этом занятии желательно присутствие родителей обучающихся (законных представителей);

- Ознакомительное занятие – педагог знакомит детей с новыми методами работы, программами, с различными наглядными материалами (учащиеся получают преимущественно теоретические знания);

- Тематическое занятие – учащимся предлагается работать над заданной темой. Занятие содействует развитию творческого воображения;

- Занятие-импровизация – на таком занятии учащиеся получают полную свободу в выборе техники для выполнения задания. Подобные занятия пробуждают фантазию ребёнка, раскрепощают его;

- Проверочное занятие – (на повторение) помогает педагогу после изучения темы проверить усвоение данного материала и выявить учащихся, которым нужна помощь педагога;

- Конкурсное занятие – строится в виде конкурсов, соревнования в игровой форме для стимулирования творчества детей;

- Занятие-экскурсия – выездные мероприятия с последующим обсуждением в объединении;

- Итоговое занятие – подводит итоги работы объединения за учебный год. Может проходить в виде просмотров творческих работ, их отбора и подготовки к отчетным выставкам.

В основу обучения положены индивидуальный и дифференцированный подходы. Каждый обучающийся определяет уровень и сферу освоения новых технологий самостоятельно.

В программе важное внимание обращается на придание процессу обучения проблемного характера, направления деятельности учащегося на самостоятельность выявления и формулирования проблемы; выработку аналитико-синтетических умений, способностей к теоретическим обобщениям. Важное место в программе отводится развитию навыков самостоятельной познавательной работы, формированию умения работать с учебными материалами, проявлению творческого подхода при выполнении самостоятельных заданий. Программа позволяет варьировать сложность материала с учетом, как возрастных особенностей развития учащихся, так и их индивидуальных проявлений.

Учащиеся работают в темпе, соответствующем их индивидуально-психологическим особенностям. Скорость работы одних обучающихся позволяет им за время занятия освоить один учебный элемент, другие успевают проработать несколько и идут с опережением, в этом случае более сильные учащиеся выступают в роли ассистентов преподавателя, консультируя и помогая более слабым. Организация взаимной помощи, совместная работа полезна всем (помогая ученик глубже осваивает материал, а другие ученики получают поддержку в решении своих проблем).

Работа по данной программе сочетает в процессе вариативной деятельности индивидуальные, групповые и коллективные формы занятий.

При этом используются разнообразные формы проведения занятий: рассказ, беседа, демонстрация с помощью обучающих программ или банка личных наработок, лекция, практическая работа, самостоятельная работа, творческие задания, конкурсы.

Наиболее предпочтительной формой обучения детей являются коллективные занятия. Следует отметить огромную важность организации педагогом на занятии обсуждения между обучающимися новой учебной темы, когда ученики беседуют не только с учителем, но и друг с другом на заданную тему. Это возможно в процессе группового поиска наилучшего варианта ответа. В этом случае педагог играет роль направляющего, но не активного участника общения.

Программа предусматривает использование следующих форм работы:

- фронтальной - подача учебного материала всему коллективу учеников;
- индивидуальной - самостоятельная работа обучающихся с оказанием помощи педагогом при возникновении затруднения, не уменьшая активности учеников и содействуя выработки навыков самостоятельной работы;
- групповой - когда учащимся предоставляется возможность самостоятельно построить свою деятельность на основе принципа

взаимозаменяемости, ощутить помощь со стороны, учесть возможности каждого на конкретном этапе деятельности.

Режим занятий, периодичность и продолжительность занятий

Общее количество часов в год на первом и втором годах обучения – 144 ч.

Занятия проходят два раза в неделю по два учебных часа. Продолжительность занятий исчисляется в академических часах – 45 минут, между занятиями установлены 10-минутные перерывы.

Объем и срок освоения программы

Срок освоения программы - 2 года

На полное освоение программы требуется - 288 часов, по 144 часа аудиторных занятий в течение одного учебного года и самоподготовки в летний период и каникулярное время.

Учебная нагрузка рассчитана на весь календарный год. В период школьных каникул внеаудиторные занятия с учащимися объединения проводятся в разных видах и формах: экскурсии в музеи города, на выставки, в парковую зону; участие в работе летней школы, летнего оздоровительного лагеря с дневным пребыванием детей, участие в организации праздников и акций, самостоятельная проектная деятельность.

Основные методы обучения

В процессе реализации программы используются различные методы обучения.

1. Методы организации и осуществления учебно-познавательной деятельности:

- словесные (рассказ; лекция; семинар; беседа; речевая инструкция; устное изложение; объяснение нового материала и способов выполнения задания; объяснение последовательности действий и содержания; обсуждение; педагогическая оценка процесса деятельности и ее результата);

- наглядные (показ видеоматериалов и иллюстраций, показ педагогом приемов программирования);

- проблемно-поисковые (создание проблемной ситуации, коллективное обсуждение, выводы);

- методы самостоятельной работы и работы под руководством педагога (создание творческих проектов);

- инструкторский метод (парное взаимодействие, более опытные учащиеся обучают менее подготовленных);

- информационные (беседа, рассказ, сообщение, объяснение, инструктаж, консультирование, анализ различных носителей информации, в

том числе Интернет сети, демонстрация, экспертиза, обзор, отчет, иллюстрация, кинопоказ);

- побудительно-оценочные (педагогическое требование и поощрение порицание и создание ситуации успеха; самостоятельная работа).

2. Методы контроля и самоконтроля за эффективностью учебно познавательной деятельности:

- устный контроль и самоконтроль (беседа, рассказ ученика, объяснение, устный опрос);

- практический контроль и самоконтроль (анализ умения работать с различными материалами);

- дидактические тесты (набор стандартизованных заданий по определенному материалу);

- наблюдения (изучение учащихся в процессе обучения).

Выбор метода обучения зависит от содержания занятий, уровня подготовленности и опыта учащихся.

Информационно-рецептивный метод применяется на теоретических занятиях.

Репродуктивный метод обучения используется на практических занятиях по отработке приёмов и навыков определенного вида работ.

Исследовательский метод применяется в работе над тематическими творческими проектами.

Для создания комфортного психологического климата на занятиях применяются следующие педагогические приёмы: создание ситуации успеха, моральная поддержка, одобрение, похвала, поощрение, доверие, доброжелательно-требовательная манера.

В ходе реализации программы используются следующие типы занятий:

- комбинированное (совмещение теоретической и практической частей занятия; проверка знаний ранее изученного материала; изложение нового материала, закрепление новых знаний, формирование умений переноса и применения знаний в новой ситуации, на практике; отработка навыков и умений с помощью решения задач);

- теоретическое (сообщение и усвоение новых знаний при объяснении новой темы, изложение нового материала, основных понятий, определение терминов, совершенствование и закрепление знаний);

- диагностическое (проводится для определения возможностей и способностей ребенка, уровня полученных знаний, умений, навыков с использованием тестирования, анкетирования, собеседования, выполнения конкурсных и творческих заданий);

- контрольное (проводится в целях контроля и проверки знаний, умений и навыков учащегося через самостоятельную и контрольную работу, индивидуальное собеседование, зачет, анализ полученных результатов. Контрольные занятия проводятся, как правило, в рамках аттестации учащихся (по пройденной теме, в начале учебного года, по окончании первого полугодия и в конце учебного года);

– практическое (является основным типом занятий, используемых в программе, как правило, содержит повторение, обобщение и усвоение полученных знаний, формирование умений и навыков, их осмысление и закрепление на практике при решении задач, использование всех видов практик.);

– вводное занятие (проводится в начале учебного года с целью знакомства с образовательной программой на год, составление индивидуальной траектории обучения; а также при введении в новую тему программы);

– итоговое занятие (проводится после изучения большой темы или раздела, по окончании полугодия, каждого учебного года и полного курса обучения).

Планируемые результаты

В результате обучения по программе «Поколение Python» обучающиеся **будут знать:**

- правила техники безопасности при работе за компьютером
- интерфейс и функционал программ редакторов кода
- планировать свою самостоятельную учебно-познавательную деятельность, выбирать индивидуальную траекторию достижения учебной цели.
- определять подходы и методы для достижения поставленной цели.
- отбирать необходимые средства для достижения поставленной цели.
- осуществлять самооценку промежуточных и итоговых результатов своей самостоятельной учебно-познавательной деятельности.
- проводить рефлексию своей учебно-познавательной деятельности.

В результате обучения по программе «Поколение Python» обучающиеся **будут уметь:**

- программировать на языке Python.
- использовать инструменты разработки среды Wing101 и /или VS Code.
- самостоятельно реализовывать проекты, связанные с разработкой игр
- работать с учебной литературой и справочными файлами, ориентироваться в информационном пространстве, анализировать, обобщать, делать выводы;
- будут уметь работать индивидуально и в команде;
- приобретут навыки творческого подхода к решению поставленных задач.
- работать с технической документацией, а также разовьют навыки поиска, обработки и анализа информации;
- разовьют навыки логического мышления и конструкторские способности;
- научатся применять изученные инструменты при выполнении научных-технических проектов;
- получают необходимые навыки для организации самостоятельной

работы;

- повысят свою информационную культуру.
- самостоятельно проектировать и выполнять сложные технические задачи;
- научатся применять изученные инструменты при самостоятельном выполнении научных-технических проектов;
- самостоятельно готовиться к участию и участвовать в чемпионатах, соревнованиях, олимпиадах различных уровней;
- организовывать проектную деятельность в направлении «Программирование»;
- создавать и отлаживать различные программы;
- получают возможность профессионального самоопределения.

Механизм оценивания образовательных результатов

Постоянно во время учебного процесса используются следующие формы контроля:

- устный опрос (в начале урока).
- тесты и викторины с использованием онлайн платформ
- тематический экзамен (в конце изучаемых тем).
- творческие задания, где усвоенный материал предлагается воплотить в оригинальные авторские проекты.

Результативность контролируется на протяжении всего процесса обучения. Для этого предусмотрены творческие задания, позволяющие проводить оценивание результатов в форме самооценки и взаимооценки. Кроме того, в конце каждой изучаемой темы проводится промежуточный контроль знаний умений и навыков.

Отследить уровень усвоения учебного материала помогает системные опросы в конце каждого занятия, тестовые задания, творческие работы учащихся, показывающие знания и умения применения инструментов при работе, конкурсы и выставки. Очень важно создать условие, чтобы каждый ученик чувствовал успешность в освоение программы, поэтому оценочная система не должна пугать и закрепощать учащихся, тесты составляются с элементами игры и юмора.

Формы подведения итогов реализации образовательной программы

Изучая программу, ребята с удовольствием реализуют свою фантазию и творческие способности. Возможность участия в конкурсах и выставках, тематических олимпиадах стимулирует творческий потенциал учащихся. Освоив азы работы с интерактивной технологией ребята осуществляют проектную деятельность на закрепление знаний и навыков.

На протяжении всего обучения каждый учащийся собирает свое «портфолио» - где хранятся все созданные работы. Это удобная форма для

демонстрации собственных навыков, творческих работ, проектов, показатель роста мастерства и просто «похвальная» папка.

В конце каждого учебного года проводится итоговая аттестация, при подведении итогов освоения программы используются:

- опрос;
- наблюдение;
- анализ, самоанализ;
- собеседование;
- тестирование и анкетирование;
- выполнение творческих заданий;
- презентации;
- викторина;
- результаты участия детей в выставках, конкурсах и фестивалях различного уровня.

Результаты аттестации отражаются в индивидуальной карте ребенка для отслеживания динамики его развития, что помогает проводить необходимую коррекцию в ходе реализации программы и конструирования учебных занятий.

Организационно-педагогические условия реализации образовательной программы

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.

Материально-техническое обеспечение

Наименование объектов и средств	Количество	Примечания
---------------------------------	------------	------------

материально-технического обеспечения		
Книгопечатная продукция (кол-во на группу)		
Образовательная программа	1 шт.	
Рабочая программа	1 шт.	
Методические разработки	15 шт.	Методические разработки занятий, оценочные материалы.
Информационно-коммуникативные средства (кол-во на группу)		
сайты	1. https://www.stepik.ru/	
	2. https://python.org/	
	3. https://wingware.com/	
	4. https://code.visualstudio.com/	
Технические средства обучения (кол-во на группу)		
Интерактивная доска	1 шт	
Проектор	1 шт	
Компьютер	15 шт	
Экранно-звуковые пособия, программное обеспечение (кол-во на группу)		
Программное обеспечение	Программное обеспечение IDE Python	распространяется бесплатно
	Программное обеспечение Wing 101	распространяется бесплатно
	Программное обеспечение Visual Studio Code	распространяется бесплатно
Электронные методические разработки	1 шт. на каждого ученика отдельно	Комплект обучающих материалов
Учебно-практическое оборудование (на группу)		
столы, парты	15 шт.	
стулья	15 шт.	
стеллажи для оборудования	6 шт.	
мебель для книг и оборудования	1 шт.	



Кадровые условия.

Педагог дополнительного образования, реализующий данную программу, должен иметь высшее профессиональное образование или среднее профессиональное образование в области, соответствующей профилю кружка, без предъявления требований к стажу работы, а также дополнительное профессиональное образование по направлению «Образование и педагогика» без предъявления требований к стажу работы.

Методическое обеспечение программы

Выбор методов и форм для реализации программы определяется:

- Поставленными целями и задачами.

- Следующими принципами обучения: от практической деятельности к внутреннему развитию всех качеств личности.
- Возможностями учащихся на данном этапе (возраст, уровень подготовки, мотивации и др.)
- Наличием соответствующей материальной базы.

Для реализации программы используются следующие *методы*:

- Развивающего обучения (проблемный, поисковый, творческий)
- Дифференцированного обучения (уровневые, индивидуальные задания)
- Игровые программы.

Одними из важнейших методов обучения по данной программе являются методы стимулирования познавательной деятельности: метод дискуссии, создание ситуации успеха.

Приоритетная технология обучения по программе – метод проектов. Обучающиеся выполняют исследовательские, творческие проекты при завершении тем, а также творческий проект, как итоговую зачетную работу, интегрирующую знания и умения по всему материалу за год.

Основной формой обучения по данной программе является учебно-практическая деятельность обучающихся. Приоритетными методами её организации служат практические, поисково-творческие работы. Все виды практической деятельности в программе направлены на освоение различных технологий работы с информацией и компьютером как инструментом обработки информации.

Дидактическое обеспечение занятий: онлайн задания, обучающие видео-уроки, учебная литература, опорные конспекты, компьютерные обучающие программы, тесты, логические схемы, различные виды раздаточных материалов, наглядные пособия, положения о конкурсах.

Оценочные и методические материалы.

Вся оценочная система делится на три уровня сложности:

- Обучающийся может ответить на общие вопросы по большинству тем, с помощью педагога может решить практическую задачу (на выбор).
- Обучающийся отвечает на все вопросы, поднимаемые за период обучения. Может самостоятельно решить практическую задачу.
- Обучающийся отвечает на все вопросы, поднимаемые за период обучения. Может самостоятельно решить любую практическую задачу в рамках курса. Также он располагает сведениями сверх программы, проявляет интерес к теме. Проявил инициативу при выполнении конкурсной работы или проекта. Вносил предложения, имеющие смысл.

Кроме того, весь курс делится на разделы. Успехи обучающегося оцениваются так же и по разделам:

- теория;
- практика;

Содержание программы
1 год обучения (144 часа, 4 часа в неделю)

1. Раздел “Ввод-вывод данных”

Тема 1. Вводное занятие. Правила техники безопасности.

Теория: Вводное занятие. Собеседование. Правила техники безопасности во время занятий в классе, с компьютерным и интерактивным оборудованием учреждения. Правила техники безопасности. Общая информация о курсе. Как проходить курс.
Форма занятия: лекция, беседа.

Тема 2. Введение в программирование

Теория: Введение в программирование. Знакомство с Python. Установка Python на компьютер. Среды разработки VS Code, Wing IDE
Практика: Выбор и установка среды программирования
Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 3. Вывод данных.

Теория: Вывод данных, команда `print()`.
Практика: Регистрация на платформе Stepik. Решение задач.
Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 4. Ввод данных.

Теория: Ввод данных, команда `input()`
Практика: Решение задач.
Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 5. Параметры команды `print()`

Теория: Необязательные параметры команды `print()`: `sep`, `end`.
Практика: Решение задач.
Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 6. Переменные.

Теория: Переменные. Стандарт PEP 8. Комментарии.
Практика: Решение задач.
Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 7. Работа с целыми числами. Часть 1.

Теория: Целые числа, основные операции с числами. Порядок выполнения операций. Преобразование типов.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 8. Работа с целыми числами. Часть 2.

Теория: Целые числа, дополнительные операции с числами. Обработка цифр числа.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 9. Итоговая работа.

Практика: Итоговая работа на ввод-вывод данных.

Форма занятия: индивидуальная работа.

2. Раздел “Условный оператор”

Тема 1. Условный оператор.

Теория: Условный оператор. Отступы. Операторы сравнения. Цепочки сравнений. Транзитивность.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 2. Логические операторы.

Теория: Логические операторы (and, or, not). Приоритет логических операций.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 3. Вложенный и каскадный условный оператор.

Теория: Вложенный и каскадный условный оператор.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 4. Итоговая работа.

Практика: Итоговая работа на условный оператор

Форма занятия: индивидуальная работа.

3. Раздел “Типы данных”

Тема 1. Числовые типы данных

Теория: Целочисленный тип данных `int`. Числа с плавающей точкой `float`. Преобразования типов данных.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 2. Встроенные функции.

Теория: Встроенные функции `max()`, `min()`, `abs()`. Функции `len()` и `str()`

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 3. Строковый тип данных.

Теория: Строковый тип данных `str`. Конкатенация строк. Умножение строки на число. Оператор `in`.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 4. Модуль `math`.

Теория: Модуль `math`, который содержит математические функции по работе с вещественными числами.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

4. Раздел “Циклы `for` и `while`”

Тема 1. Цикл `for`. Часть 1.

Теория: Цикл `for`. Переменная цикла.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 2. Цикл `for`. Часть 2.

Теория: Цикл `for`. Функции `range()` с одним параметром. Функции `range()` с двумя параметрами. Функции `range()` с тремя параметрами.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 3. Частые сценарии.

Теория: Частые сценарии при написании циклов. Подсчет количества. Вычисление суммы и произведения. Обмен значений переменных. Сигнальные метки. Определение максимума и минимума. Расширенные операторы присваивания.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 4. Цикл while. Часть 1.

Теория: Цикл while. Считывание данных до стоп значения. Бесконечный цикл.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 5. Цикл while. Часть 2.

Теория: Использование цикла while для обработки цифр числа.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 6. Операторы break, continue

Теория: Оператор break. Оператор continue. Бесконечные циклы. Ключевое слово else в циклах.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 7. Ревью кода.

Теория: Ревью кода. Поиск ошибок. Производительность кода.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 8. Вложенные циклы.

Теория: Вложенные циклы. Операторы break и continue во вложенных циклах.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 9. Итоговая работа.

Практика: Итоговая работа на циклы.

Форма занятия: индивидуальная работа.

5. Раздел “Строковый тип данных”

Тема 1. Индексация.

Теория: Индексация строк. Итерирование строк.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 2. Срезы.

Теория: Срезы строк. Изменение символов строки.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 3. Методы строк. Часть 1

Теория: Методы строк `capitalize()`, `swapcase()`, `title()`, `lower()`, `upper()`.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 4. Методы строк. Часть 2

Теория: Методы строк `count()`, `startswith()`, `endswith()`, `find()`, `rfind()`, `index()`, `rindex()`, `strip()`, `lstrip()`, `rstrip()`, `replace()`.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 5. Методы строк. Часть 3

Теория: Методы строк `isalnum()`, `isalpha()`, `isdigit()`, `islower()`, `isupper()`, `isspace()`.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 6. Форматирование строк.

Теория: Метод `format()`. f-строки.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 7. Таблицы символов.

Теория: Представление строк в памяти компьютера. Таблица символов ASCII. Таблица символов Unicode. Функция `ord()`. Функция `chr()`.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 8. Сравнение строк.

Теория: Сравнение строк единичной длины, сравнение строк не единичной длины.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 9. Итоговая работа.

Практика: Итоговая работа на строки

Форма занятия: индивидуальная работа.

6. Раздел “Списки”

Тема 1. Введение в списки.

Теория: Создание списков. Пустые списки. Встроенная функция `list()`. Вывод списков.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 2. Основы работы со списками.

Теория: Встроенные функции `len()`, `sum()`, `min()`, `max()`. Оператор принадлежности `in`. Индексация и срезы. Конкатенация и умножение на число. Отличие списков от строк.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 3. Методы списков. Часть 1.

Теория: Метод добавления элемента `append()`. Метод расширения списка `extend()`. Оператор `del`.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 4. Элементы списка.

Теория: Вывод элементов списка с помощью цикла `for`. Вывод элементов списка с помощью распаковки.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 5. Методы списков. Часть 2.

Теория: Методы `split()` и `join()`.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 6. Методы списков. Часть 3.

Теория: Методы `insert()`, `index()`, `remove()`, `pop()`, `reverse()`, `count()`, `clear()`, `copy()`, `sort()`.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 7. Списочные выражения.

Теория: Списочные выражения.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 8. Сортировка.

Теория: Задача сортировки. Алгоритмы сортировки. Сортировка пузырьком. Сортировка выбором. Сортировка простыми вставками.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 9. Итоговая работа.

Практика: Итоговая работа на списки.

Форма занятия: индивидуальная работа.

7. Раздел “Функции”

Тема 1. Функции без параметров.

Теория: Функции без параметров. Преимущества использования функций. Ключевое слово `pass`.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 2. Функции с параметрами.

Теория: Функции с параметрами. Область видимости параметрической переменной. Параметры по умолчанию.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 3. Локальные и глобальные переменные.

Теория: Локальные и глобальные переменные, их области видимости, глобальные константы.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 4. Функции с возвратом значения. Часть 1.

Теория: Функции с возвратом значения. Использование инструкции return.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 5. Функции с возвратом значения. Часть 2.

Теория: Функции с возвратом значения. Возвращение булевых значений.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 6. Функции с возвратом значений. Часть 3.

Теория: Функции с возвратом нескольких значений.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 7. Итоговая работа.

Практика: Итоговая работа на функции.

Форма занятия: индивидуальная работа.

8. Раздел “Работа над мини-проектами”

Тема 1. Случайные числа.

Практика: Работа над проектом “Случайные числа” с использованием модуля random.

Форма занятия: индивидуальная работа и в малых группах.

Тема 2. Числовая угадайка.

Практика: Работа над проектом “Числовая угадайка.”

Форма занятия: индивидуальная работа и в малых группах.

Тема 3. Магический шар 8.

Практика: Работа над проектом “Магический шар 8.”

Форма занятия: индивидуальная работа и в малых группах.

Тема 4. Генератор безопасных паролей.

Практика: Работа над проектом “Генератор безопасных паролей.”

Форма занятия: индивидуальная работа и в малых группах.

Тема 5. Шифр Цезаря.

Практика: Работа над проектом “Шифр Цезаря.”

Форма занятия: индивидуальная работа и в малых группах.

Тема 6. Калькулятор систем счисления.

Практика: Работа над проектом “Калькулятор систем счисления.”

Форма занятия: индивидуальная работа и в малых группах.

Тема 7. Угадайка слов.

Практика: Работа над проектом “Угадайка слов.”

Форма занятия: индивидуальная работа и в малых группах.

Тема 8. Заключительный урок.

Теория: Заключительный урок. Подведение итогов.

Форма занятия: беседа, обмен мнениями.

**Содержание программы**

2 год обучения (144 часа, 4 часа в неделю)

1. Раздел “Повторяем основные конструкции языка Python”**Тема 1. Вводное занятие. Правила техники безопасности.**

Теория: Вводное занятие. Собеседование. Правила техники безопасности во время занятий в классе, с компьютерным и интерактивным оборудованием учреждения. Правила техники безопасности. Общая информация о курсе. Как проходить курс.

Форма занятия: лекция, беседа.

Тема 2. Повторяем.

Практика: Повторяем ввод-вывод данных, типы данных, условный оператор, циклы. Решение задач.

Форма занятия: беседа, индивидуальная работа и в малых группах.

Тема 3. Повторяем.

Практика: Повторяем строки, списки, функции. Решение задач.

Форма занятия: беседа, индивидуальная работа и в малых группах.

2. Раздел “Тип данных bool и NoneType”

Тема 1. Типы данных bool, Nonetype.

Теория: Типы данных bool, Nonetype.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

3. Раздел “Вложенные списки, матрицы”

Тема 1. Повторяем списки.

Теория: Повторяем списки. Объявление, индексация, срезы, перебор элементов, методы списков.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 2. Вложенные списки. Часть 1.

Теория: Вложенные списки, объявление и индексация, функции len(), max(), min(), списочные методы.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 3. Вложенные списки. Часть 2.

Теория: Создание, считывание, перебор элементов, обработка, вывод вложенных списков.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 4. Матрицы. Часть 1.

Теория: Работа с матрицами, квадратные и прямоугольные матрицы, функции ljust() и rjust(), главная и побочная диагонали.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 5. Матрицы. Часть 2.

Теория: Симметричная матрица, обмен столбцов, обмен диагоналей, поворот матрицы.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 6. Матрицы. Часть 3.

Теория: Заполнение матрицы змейкой, заполнение диагоналями, заполнение спиралью.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 7. Операции над матрицами.

Теория: Операции над матрицами в математике. Сложение матриц, умножение матрицы на число, умножение матриц

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 8. Итоговая работа.

Практика: Итоговая работа на матрицы.

Форма занятия: индивидуальная работа.

4. Раздел “Кортежи”

Тема 1. Введение в кортежи.

Теория: Тип данных tuple. Особенности работы с кортежами.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 2. Основы работы с кортежами. Часть 1.

Теория: Функция tuple(). Особенности кортежей. Методы кортежей.

Вложенные кортежи.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 3. Основы работы с кортежами. Часть 2.

Теория: Перебор кортежей. Сравнение кортежей. Сортировка кортежей.

Преобразование кортежа в список и строку. Упаковка кортежей.

Распаковка кортежей. Присваивание кортежей.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 4. Итоговая работа.

Практика: Итоговая работа на кортежи.

Форма занятия: индивидуальная работа.

5. Раздел “Множества”

Тема 1. Множества в математике.

Теория: Множества в математике. Числовые множества.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 2. Операции над множествами.

Теория: Операции над множествами. Диаграммы Эйлера-Вена.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 3. Введение в множества в Python.

Теория: Введение в множества в Python. Создание множеств. Пустые множества. Встроенная функция `set()`. Вывод множеств.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 4. Основы работы с множествами.

Теория: Встроенные функции `len()`, `sum()`, `min()`, `max()`. Оператор принадлежности `in`. Перебор множеств. Форматированный вывод множеств. Сравнение множеств.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 5. Методы множеств. Часть 1.

Теория: Метод добавления элемента `add()`. Методы удаления элементов `remove()`, `discard()`, `pop()`. Метод удаления всех элементов `clear()`.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 6. Методы множеств. Часть 2.

Теория: Методы `union()`, `intersection()`, `difference()`, `symmetric_difference()`. Методы `update()`, `intersection_update()`, `difference_update()`, `symmetric_difference_update()`. Операторы `&`, `|`, `-`, `^`.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 7. Методы множеств. Часть 3.

Теория: Методы `issuperset()`, `issubset()`, `isdisjoint()`. Сравнение множеств (`<`, `>`, `<=`, `>=`).

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 8. Генераторы множеств.

Теория: Генераторы множеств. Неизменяемые множества `frozenset`.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 9. Итоговая работа.

Практика: Итоговая работа на множества.

Форма занятия: индивидуальная работа.

6. Раздел “Словари”

Тема 1. Введение в словари в Python.

Теория: Введение в словари в Python. Создание словарей. Обращение по ключу. Встроенная функция `dict()`. Создание словарей на основе списков и кортежей. Пустой словарь. Вывод словаря. Особенности словарей.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 2. Основы работы со словарями.

Теория: Основы работы со словарями. Встроенные функции `len()`, `sum()`, `min()`, `max()`. Оператор принадлежности `in`. Перебор словарей. Распаковка словаря. Форматированный вывод словарей. Сравнение словарей. Методы `keys()`, `values()` и `items()`.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 3. Методы словарей.

Теория: Методы словарей. Добавление и изменение элементов в словаре. Удаление элементов из словаря. Методы `get()`, `update()`, `pop()`, `popitem()`, `clear()`, `copy()`, `setdefault()`.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 4. Типы задач, где используются словари.

Теория: Задачи на словари.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 5. Вложенные словари и генераторы словарей.

Теория: Вложенные словари и генераторы словарей.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 6. Итоговая работа.

Практика: Итоговая работа на словари.

Форма занятия: индивидуальная работа.

7. Раздел “Модули random и string”

Тема 1. Модуль random. Часть 1.

Теория: Случайные числа. Псевдослучайные числа.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 2. Модуль random. Часть 2.

Теория: Метод `shuffle()`. Метод `choice()`. Метод `sample()`. Модуль `string`.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 3. Метод Монте-Карло и Bogosort.

Теория: Метод Монте-Карло и Bogosort.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

8. Раздел “Модули decimal, fractions и complex”

Тема 1. Модуль decimal.

Теория: Модуль `decimal`. Числа с плавающей точкой `float`. Тип данных `Decimal`.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 2. Модуль fractions.

Теория: Модуль `fractions`. Тип данных `Fraction`.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 3. Тип данных complex.

Теория: Модуль cmath. Тип данных complex. Вещественные числа. Мнимые числа. Комплексные числа. Тип данных complex.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

9. Раздел “Модуль turtle”

Тема 1. Модуль turtle (черепашки). Часть 1.

Теория: Черепаший графика. Модуль turtle. Рисование отрезков прямой. Поворот черепашки. Установка углового направления черепашки. Изменение внешнего вида черепашки.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 2. Модуль turtle (черепашки). Часть 2.

Теория: Поднятие и опускание пера. Рисование кругов и точек. Изменение размера пера, цвета рисунка, цвета фона. Создание штампа. Возвращение экрана в исходное состояние. Получение текущей позиции черепашки. Управление скоростью анимации черепашки.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 3. Модуль turtle (черепашки). Часть 3.

Теория: Вывод текста в графическое окно. Заполнение геометрических фигур. Создание нескольких черепашек. Отслеживание нажатия клавиш. Отслеживание нажатия мыши.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

10. Раздел “Функции”

Тема 1. Функции. Необязательные и именованные аргументы.

Теория: Функции. Позиционные аргументы. Необязательные аргументы. Именованные аргументы.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 2. Функции с переменным количеством аргументов.

Теория: Функции с переменным количеством аргументов. Аргументы `*args`, аргументы `**kwargs`. Keyword-only аргументы.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 3. Парадигмы программирования.

Теория: Императивное программирование. Структурное программирование. Объектно-ориентированное программирование. Логическое программирование. Функциональное программирование.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 4. Функции как объекты.

Теория: Функции в качестве аргументов других функций. Встроенные функции, принимающие функции в качестве аргументов. Функции в качестве возвращаемых значений других функций.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 5. Функции высшего порядка.

Теория: Функции высшего порядка. Самописные функции `map()`, `filter()`, `reduce()`.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 6. Встроенные функции `map()`, `filter()`, `reduce()`.

Теория: Встроенные функции `map()`, `filter()`, `reduce()`. Модуль `operator`.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 7. Анонимные функции. Часть 1.

Теория: Определение анонимных функций. Условный оператор в теле анонимной функции.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 8. Анонимные функции. Часть 2.

Теория: Передача аргументов в анонимную функцию. Ограничения анонимных функций.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 9. Встроенные функции `any()`, `all()`, `zip()`, `enumerate()`.

Теория: Встроенные функции any(), all(), zip(), enumerate().

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 10. Итоговая работа.

Практика: Итоговая работа на функции.

Форма занятия: индивидуальная работа.

11. Раздел “Работа с файлами”

Тема 1. Файловый ввод и вывод.

Теория: Файловый ввод и вывод. Понятие файла. Файловый ввод и вывод. Работа с файлами. Типы файлов. Методы доступа к файлам. Имена файлов. Кодировка файлов.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 2. Работа с текстовыми файлами. Часть 1.

Теория: Открытие и закрытие файлов. Указание кодировки. Чтение данных из файла.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 3. Работа с текстовыми файлами. Часть 2.

Теория: Позиция (курсор) в файле. Менеджеры контекста.

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 4. Работа с текстовыми файлами. Часть 3.

Теория: Запись данных в файл. Файловые методы write() и writelines().

Практика: Решение задач.

Форма занятия: лекция, беседа, индивидуальная работа и в малых группах.

Тема 5. Итоговая работа.

Практика: Итоговая работа на файлы.

Форма занятия: индивидуальная работа.

Тема 6. Заключительный урок.

Теория: Заключительный урок. Подведение итогов.

Форма занятия: беседа, обмен мнениями.

БАЗОВЫЙ УРОВЕНЬ
1-й год обучения (144 часа, 4 часа в неделю)

Программа первого года предполагает постепенное знакомство учащихся с основами языка Python.

В начале ученики знакомятся с различными простыми типами данных (числа, строки, списки, и т.д.) и работой с ними (арифметические, логические операции, ввод-вывод данных в программу и т.д.), изучают методы для работы с ними, а также осваивают базовые конструкции имеющиеся в любом языке программирования - операторы ветвления, циклы, функции.

На курсе очень много практики, так как в памяти надежно сохраняется только то, что мы подкрепляем собственными действиями, опытом, так устроен наш мозг.

Задачи первого года обучения.

Образовательные:

- дать представление о языке высокого уровня Python;
- научить логически и алгоритмически думать;
- научить составлять алгоритмы для решения задач;
- формировать навыки культуры написания кода;
- научить работе с онлайн платформой;
- научить базовой работе в редакторах кода;

Развивающие:

- развитие навыков для работы с учебной онлайн платформой;
- развитие навыков работать с учебной литературой;
- развитие навыков работать со справочными данными;
- развитие логического, образного и абстрактного мышления;
- развитие интереса к программированию.

Воспитательные:

- прививать интерес к занятиям;
- создать условия для личностного развития учащихся;
- создать условия для позитивной социализации;
- создать условия для профессионального самоопределения;
- формирование культуры здорового и безопасного образа жизни.

Учебный план
(1 год обучения)

№	Название темы	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
1. Раздел “Ввод-вывод данных”					
1	Вводное занятие. Правила техники	2	2	0	

	безопасности.Общая информация о курсе. Как проходить курс.				
2	- Введение в программирование. Знакомство с Python. Установка Python на компьютер. Среда разработки - VS Code. Среда разработки Wing IDE	2	1	1	
3	Вывод данных, команда <code>print()</code> . Решение задач.	2	0,5	1,5	
4	Ввод данных, команда <code>input()</code> . Решение задач.	2	0,5	1,5	
5	Необязательные параметры команды <code>print()</code> : <code>sep</code> , <code>end</code> . Решение задач.	2	0,5	1,5	
6	Переменные. Стандарт PEP 8. Комментарии Решение задач	2	0,5	1,5	
7	Целые числа, основные операции с числами. Порядок выполнения операций. Преобразование типов. Решение задач.	2	0,5	1,5	
8	Целые числа, дополнительные операции с числами. Обработка цифр числа. Решение задач.	4	1	3	
9	Итоговая работа на ввод-вывод данных	2	0	2	контрольная работа
2. Раздел “Условный оператор”					
1	Условный оператор. Отступы. Операторы сравнения. Цепочки сравнений. Транзитивность. Решение задач.	2	0,5	1,5	
2	Логические операторы (<code>and</code> , <code>or</code> , <code>not</code>). Приоритет логических операций. Решение задач.	2	0,5	1,5	
3	Вложенный и каскадный условный оператор. Решение задач.	2	0,5	1,5	
4	Итоговая работа на условный оператор	2	0	2	контрольная работа
3. Раздел “Типы данных”					
1	Целочисленный тип данных <code>int</code> . Числа с плавающей точкой <code>float</code> . Преобразования типов данных. Решение задач.	2	0,5	1,5	
2	Встроенные функции <code>max()</code> , <code>min()</code> , <code>abs()</code> . Функции <code>len()</code> и <code>str()</code> . Решение задач.	2	0,5	1,5	
3	Строковый тип данных <code>str</code> . Конкатенация строк. Умножение строки на число. Оператор <code>in</code> . Решение задач	2	0,5	1,5	
4	Модуль <code>math</code> . Решение задач.	2	0,5	1,5	
4. Раздел “Циклы <code>for</code> и <code>while</code>”					
1	Цикл <code>for</code> . Переменная цикла. Решение задач.	4	1	3	
2	Цикл <code>for</code> . Функции <code>range()</code> с одним параметром. Функции <code>range()</code> с двумя	4	1	3	

	параметрами. Функции range() с тремя параметрами. Решение задач.				
3	Частые сценарии при написании циклов. Подсчет количества. Вычисление суммы и произведения. Обмен значений переменных. Сигнальные метки. Определение максимума и минимума. Расширенные операторы присваивания. Решение задач.	4	1	3	
4	Цикл while. Считывание данных до стоп значения. Бесконечный цикл. Решение задач.	2	0,5	1,5	
5	Использование цикла while для обработки цифр числа. Решение задач.	4	1	3	
6	Оператор break. Оператор continue. Бесконечные циклы. Ключевое слово else в циклах. Решение задач.	2	0,5	1,5	
7	Ревью кода. Поиск ошибок. Производительность кода. Решение задач.	2	0,5	1,5	
8	Вложенные циклы. Операторы break и continue во вложенных циклах. Решение задач	4	1	3	
9	Итоговая работа на циклы	2	0	2	контрольная работа
5. Раздел “Строковый тип данных”					
1	Индексация строк. Итерирование строк. Решение задач.	2	0,5	1,5	
2	Срезы строк. Изменение символов строки. Решение задач.	4	1	3	
3	Методы строк. Часть 1. Решение задач.	2	0,5	1,5	
4	Методы строк. Часть 2. Решение задач	2	0,5	1,5	
5	Методы строк. Часть 3. Решение задач.	2	0,5	1,5	
6	Форматирование строк	2	0,5	1,5	
7	Представление строк в памяти компьютера. Таблица символов ASCII. Таблица символов Unicode. Функция ord(). Функция chr(). Решение задач.	2	0,5	1,5	
8	Сравнение строк	2	0,5	1,5	
9	Итоговая работа на строки	2	0	2	контрольная работа
6. Раздел “Списки”					
1	Создание списков. Пустые списки. Встроенная функция list(). Вывод списков.Решение задач.	2	0,5	1,5	
2	Встроенные функции len(),sum(),min(),max(). Оператор принадлежности in. Индексация и срезы. Конкатенация и умножение на число. Отличие списков от строк. Решение задач.	2	0,5	1,5	
3	Метод добавления элемента append(). Метод расширения списка extend().	4	1	3	

	Оператор del. Решение задач.				
4	Вывод элементов списка с помощью цикла for. Вывод элементов списка с помощью распаковки. Решение задач.	2	0,5	1,5	
5	Методы split(), join(). Решение задач.	2	0,5	1,5	
6	Методы списков insert(), index(), pop(), remove(), reverse(), count(), clear(), copy(), sort(). Решение задач.	4	1	3	
7	Списочные выражения. Решение задач.	4	1	3	
8	Задача сортировки. Алгоритмы сортировки (пузырьком, выбором, простыми вставками). Решение задач.	2	0,5	1,5	
9	Итоговая работа на списки	2	0	2	контрольная работа
7. Раздел “Функции”					
1	Функции без параметров. Преимущества использования функций. Ключевое слово pass. Решение задач.	2	0,5	1,5	
2	Функции с параметрами. Область видимости параметрической переменной. Параметры по умолчанию. Решение задач.	2	0,5	1,5	
3	Локальные переменные. Область действия локальной переменной. Глобальные переменные. Глобальные константы. Решение задач.	2	0,5	1,5	
4	Функции с возвратом значения. Часть 1. Использование инструкции return. Решение задач.	4	1	3	
5	Функции с возвратом значения. Часть 2.  Возвращение булевых значений. Решение задач.	4	1	3	
6	Функции с возвратом нескольких значений. Решение задач.	4	1	3	
7	Итоговая работа на функции	2	0	2	контрольная работа
8. Раздел “Работа над мини-проектами”					
1	Случайные числа. Модуль random.	2	0	2	зачет
2	Числовая угадка.	2	0	2	зачет
3	Магический шар 8.	2	0	2	зачет
4	Генератор безопасных паролей.	2	0	2	зачет
5	Шифр Цезаря.	2	0	2	зачет
6	Калькулятор систем счисления.	2	0	2	зачет
7	Угадайка слов.	2	0	2	зачет
8	Заключительный урок. Подведение итогов.	2	2	0	
	Итого	144	33	111	

БАЗОВЫЙ УРОВЕНЬ

2-й год обучения (144 часа, 4 часа в неделю)

Программа второго года предполагает продолжение знакомства учащихся с основами языка Python, опираясь на знания и умения полученные от изучения программы первого года обучения.

Все начинается с краткого повторения уже полученных знаний, потом переходим к изучению некоторых новых типов данных (вложенные списки, матрицы, кортежи, множества, словари), методов для работы с ними, библиотек и модулей (random, string, decimal, fractions, complex, turtle) и заканчиваем изучением работы с файлами и продвинутыми функциями (функции с переменным количеством аргументов, функции высшего порядка, анонимные функции, различные встроенные функции)

Все также на курсе очень много практики, так как в памяти надежно сохраняется только то, что мы подкрепляем собственными действиями, опытом, так уж устроен наш мозг.

Задачи второго года обучения.

Образовательные:

- познакомить с различными технологиями, применяемых при программировании;
- научить составлять алгоритмы для решения задач;
- научить оптимизации алгоритмов и программ;
- формировать навыки культуры написания кода;
- научить практической работе в редакторах кода;

Развивающие:

- развитие навыков анализировать, обобщать, делать выводы;
- развитие навыков по поиску и работы с новой информацией;
- развивает умение искать и исправлять ошибки;
- развитие логического, образного и абстрактного мышления;
- развитие умения работать индивидуально и в команде;

Воспитательные:

- воспитывать упорство в достижении желаемого результата;
- воспитывать эстетический вкус;
- воспитывать чувство взаимопомощи, доверия, коллективизма;
- воспитание трудолюбия;
- воспитание культуры здорового и безопасного образа жизни.

Учебный план

(2 год обучения)

№	Название темы	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	

1. Раздел “Повторяем основные конструкции языка Python”					
1	Вводное занятие. Правила техники безопасности. Общая информация о курсе. Как проходить курс.	2	2	0	
2	• Повторяем ввод-вывод данных, типы данных, условный оператор, циклы. Решение задач.	2	0	2	
3	• Повторяем строки, списки, функции. Решение задач.	2	0	2	
2. Раздел “Тип данных bool и NoneType”					
1	Типы данных bool, Nonetype. Решение задач.	2	0,5	1,5	
3. Раздел “Вложенные списки, матрицы”					
1	Повторяем списки. Решение задач.	2	0,5	1,5	
2	Вложенные списки. Часть 1 Вложенные списки, объявление и индексация, функции len(), max(), min(), списочные методы. Решение задач.	4	1	3	
3	Вложенные списки. Часть 2 Создание, считывание, перебор элементов, обработка, вывод вложенных списков. Решение задач.	4	1	3	
4	Матрицы. Часть 1 Работа с матрицами, квадратные и прямоугольные матрицы, функции ljust() и rjust(), главная и побочная диагонали. Решение задач.	4	1	3	
5	Матрицы. Часть 2 Симметричная матрица, обмен столбцов, обмен диагоналей, поворот матрицы. Решение задач.	4	1	3	
6	Матрицы. Часть 3 Заполнение матрицы змейкой, заполнение диагоналями, заполнение спиралью. Решение задач.	4	1	3	
7	Операции над матрицами в математике. Сложение матриц, умножение матрицы на число, умножение матриц. Решение задач.	2	0,5	1,5	
8	Итоговая работа на матрицы.	2	0	2	контрольная работа
4. Раздел “Кортежи”					
1	Введение в кортежи. Тип данных tuple. Особенности работы с кортежами. Решение задач.	2	0,5	1,5	
2	Основы работы с кортежами. Часть 1. Функция tuple(). Особенности кортежей. Методы кортежей. Вложенные кортежи. Решение задач.	2	0,5	1,5	
3	Основы работы с кортежами. Часть 2.	4	1	3	

	Перебор кортежей. Сравнение кортежей. Сортировка кортежей. Преобразование кортежа в список и строку. Упаковка кортежей. Распаковка кортежей. Присваивание кортежей. Решение задач.				
4	Итоговая работа на кортежи.	2	0	2	контрольная работа
5. Раздел “Множества”					
1	Множества в математике. Числовые множества. Решение задач.	2	0,5	1,5	
2	Операции над множествами. Диаграммы Эйлера-Вена. Решение задач.	2	0,5	1,5	
3	Введение в множества в Python. Создание множеств. Пустые множества. Встроенная функция set(). Вывод множеств. Решение задач.	2	0,5	1,5	
4	Основы работы с множествами. Встроенные функции len(), sum(), min(), max(). Оператор принадлежности in. Перебор множеств. Форматированный вывод множеств. Сравнение множеств. Решение задач.	4	1	3	
5	Методы множеств. Часть 1. Метод добавления элемента add(). Методы удаления элементов remove(), discard(), pop(). Метод удаления всех элементов clear(). Решение задач.	2	0,5	1,5	
6	Методы множеств. Часть 2. Методы union(), intersection(), difference(), symmetric_difference(). Методы update(), intersection_update(), difference_update(), symmetric_difference_update(). Операторы &, , -, ^. Решение задач.	4	1	3	
7	Методы множеств. Часть 3. Методы issuperset(), issubset(), isdisjoint(). Сравнение множеств (<, >, <=, >=). Решение задач.	4	1	3	
8	Генераторы множеств. Неизменяемые множества frozenset. Решение задач.	2	0,5	1,5	
9	Итоговая работа на множества.	2	0	2	контрольная работа
6. Раздел “Словари”					
1	Введение в словари в Python. Создание словарей. Обращение по ключу. Встроенная функция dict(). Создание словарей на основе списков и кортежей. Пустой словарь. Вывод словаря. Особенности словарей. Решение задач.	2	0,5	1,5	
2	Основы работы со словарями. Встроенные функции len(), sum(), min(),	2	0,5	1,5	

	max(). Оператор принадлежности in. Перебор словарей. Распаковка словаря. Форматированный вывод словарей. Сравнение словарей. Методы keys(), values() и items(). Решение задач.				
3	Методы словарей. Добавление и изменение элементов в словаре. Удаление элементов из словаря. Методы get(), update(), pop(), popitem(), clear(), copy(), setdefault(). Решение задач.	4	1	3	
4	Задачи на словари. Решение задач.	2	0,5	1,5	
5	Вложенные словари и генераторы словарей. Решение задач.	2	0,5	1,5	
6	Итоговая работа на словари.	2	0	2	контрольная работа
7. Раздел “Модули random и string”					
1	Модуль random. Часть 1. Случайные числа. Псевдослучайные числа. Решение задач.	2	0,5	1,5	
2	Модуль random. Часть 2. Метод shuffle(). Метод choice(). Метод sample(). Модуль string. Решение задач.	2	0,5	1,5	
3	Метод Монте-Карло и Bogosort. Решение задач.	2	0,5	1,5	
8. Раздел “Модули decimal, fractions и complex”					
1	Модуль decimal. Числа с плавающей точкой float. Тип данных Decimal. Решение задач.	2	0,5	1,5	
2	Модуль fractions. Тип данных Fraction. Решение задач.	2	0,5	1,5	
3	Тип данных complex. Вещественные числа. Мнимые числа. Комплексные числа. Тип данных complex. Модуль cmath. Решение задач.	2	0,5	1,5	
9. Раздел “Модуль turtle”					
1	Модуль черепашки. Часть 1. Черепашка графика. Модуль turtle. Рисование отрезков прямой. Поворот черепашки. Установка углового направления черепашки. Изменение внешнего вида черепашки. Решение задач.	2	0,5	1,5	
2	Модуль черепашки. Часть 2. Поднятие и опускание пера. Рисование кругов и точек. Изменение размера пера, цвета рисунка, цвета фона. Создание штампа. Возвращение экрана в исходное состояние. Получение текущей позиции черепашки. Управление скоростью анимации черепашки. Решение задач.	2	0,5	1,5	
3	Модуль черепашки. Часть 3. Вывод	2	0,5	1,5	

	текста в графическое окно. Заполнение геометрических фигур. Создание нескольких черепашек. Отслеживание нажатия клавиш. Отслеживание нажатия мыши. Решение задач.				
10. Раздел “Функции”					
1	Функции. Позиционные аргументы. Необязательные аргументы. Именованные аргументы. Решение задач.	2	0,5	1,5	
2	Функции с переменным количеством аргументов. Аргументы *args, аргументы **kwargs. Keyword-only аргументы. Решение задач.	4	1	3	
3	Парадигмы программирования. Императивное программирование. Структурное программирование. Объектно-ориентированное программирование. Логическое программирование. Функциональное программирование. Решение задач.	2	0,5	1,5	
4	Функции как объекты. Функции в качестве аргументов других функций. Встроенные функции, принимающие функции в качестве аргументов. Функции в качестве возвращаемых значений других функций. Решение задач.	2	0,5	1,5	
5	Функции высшего порядка. Самописные функции map(), filter(), reduce(). Решение задач.	4	1	3	
6	Встроенные функции map(), filter(), reduce(). Модуль operator. Решение задач.	4	1	3	
7	Анонимные функции. Часть 1. Определение анонимных функций. Условный оператор в теле анонимной функции. Решение задач.	2	0,5	1,5	
8	Анонимные функции. Часть 2. Передача аргументов в анонимную функцию. Ограничения анонимных функций. Решение задач.	2	0,5	1,5	
9	Встроенные функции any(), all(), zip(), enumerate(). Решение задач.	4	1	3	
10	Итоговая работа на функции.	2	0	2	контрольная работа
11. Раздел “Работа с файлами”					
1	Файловый ввод и вывод. Понятие файла. Файловый ввод и вывод. Работа с файлами. Типы файлов. Методы доступа к файлам. Имена файлов. Кодировка файлов. Решение задач.	2	0,5	1,5	

2	Работа с текстовыми файлами. Часть 1. Открытие и закрытие файлов. Указание кодировки. Чтение данных из файла. Решение задач.	4	1	3	
3	Работа с текстовыми файлами. Часть 2. Позиция (курсор) в файле. Менеджеры контекста. Решение задач.	4	1	3	
4	Работа с текстовыми файлами. Часть 3. Запись данных в файл. Файловые методы write() и writelines(). Решение задач.	2	0,5	1,5	
5	Итоговая работа на файлы.	2	0	2	контрольная работа
6	Заключительный урок. Подведение итогов.	2	2	0	
	Итого	144	35	109	

Календарный учебный график разноуровневой дополнительной общеразвивающей программы «Поколение Python»

1-й год обучения

Режим деятельности	Разноуровневая дополнительная общеобразовательная общеразвивающая программа «Поколение Python»
Начало учебного года	15 сентября
Продолжительность учебного периода	36 учебных недель
Продолжительность учебной недели	5 дней
Периодичность учебных занятий	2 раза в неделю по 2 учебных часа
Количество часов	144 часа
Окончание учебного года	31 мая
Период реализации программы	15.09.2025-31.05.2026

2-й год обучения

Режим деятельности	Разноуровневая дополнительная общеобразовательная общеразвивающая программа «Поколение Python»
Начало учебного года	15 сентября
Продолжительность учебного периода	36 учебных недель

Продолжительность учебной недели	5 дней
Периодичность учебных занятий	2 раза в неделю по 2 учебных часа
Количество часов	144 часа
Окончание учебного года	31 мая
Период реализации программы	15.09.2025-31.05.2026

Рабочая программа воспитания

Воспитательный компонент осуществляется по следующим направлениям организации воспитания и социализации обучающихся:

- 1) гражданско-патриотическое
- 2) нравственное и духовное воспитание;
- 3) воспитание положительного отношения к труду и творчеству;
- 4) интеллектуальное воспитание;
- 5) здоровьесберегающее воспитание;
- 6) правовое воспитание и культура безопасности;
- 7) воспитание семейных ценностей;
- 8) формирование коммуникативной культуры;
- 9) экологическое воспитание.

Цель – формирование гармоничной личности с широким мировоззренческим кругозором, с серьезным багажом теоретических знаний и практических навыков, посредством информационно-коммуникативных технологий.

Используемые формы воспитательной работы: викторина, экскурсии, игровые программы, диспуты.

Методы: беседа, мини-викторина, моделирование, наблюдения, столкновения взглядов и позиций, проектный, поисковый.

Планируемый результат: повышение мотивации к изобретательству и созданию собственных конструкций; сформированность настойчивости в достижении цели, стремление к получению качественного законченного результата; умение работать в команде; сформированность нравственного, познавательного и коммуникативного потенциалов личности.

Календарный план воспитательной работы

№ п/п	Название мероприятия, события	Направления воспитательной работы	Форма проведения	Сроки проведения
1.	Инструктаж по технике безопасности при работе с компьютером, правила поведения на занятиях.	Безопасность и здоровый образ жизни	В рамках занятий	Сентябрь

2.	Игры на знакомство и командообразование	Нравственное воспитание	В рамках занятий	Сентябрь
3.	Беседа о сохранении материальных ценностей, бережном отношении к оборудованию	Нравственное воспитание, трудовое воспитание	В рамках занятий	Сентябрь-май
4.	Беседа о празднике «День народного единства»	Гражданско-патриотическое воспитание, нравственное воспитание	В рамках занятий	Ноябрь
5.	Участие в соревнованиях и конкурсах, различного уровня	Воспитание интеллектуально познавательных интересов	В рамках занятий	Ноябрь-май
6.	Инструктаж на тему “Безопасное поведение в зимний период”	Безопасность и здоровый образ жизни	В рамках занятий	Декабрь
7.	Беседа о празднике «День защитника Отечества»	Гражданско-патриотическое, нравственное и духовное воспитание; воспитание семейных ценностей	В рамках занятий	Февраль
8.	Беседа о «Штурме Кенигсберга»	Гражданско-патриотическое, нравственное и духовное воспитание; воспитание семейных ценностей	В рамках занятий	Апрель
9.	Беседа о празднике «День Победы»	Гражданско-патриотическое, нравственное и духовное воспитание; воспитание семейных ценностей	В рамках занятий	Май
10.	Инструктаж на тему “Безопасное поведение в летний период”	Безопасность и здоровый образ жизни	В рамках занятий	Май

Список использованной литературы:

Нормативные правовые акты:

1. Конституция Российской Федерации. (принята всенародным голосованием 12 декабря 1993 года с изменениями, одобренными в ходе общероссийского голосования 1 июля 2020 года.)[Электронный ресурс]
Режим доступа: <http://pravo.gov.ru/constitution/>

2. Конвенция о правах ребенка: [принята резолюцией № 44/25 Генеральной Ассамблеи ООН 20 ноября 1989г., вступила в силу в СССР 15 сентября 1990 г.]. – М.: Омега-Л, 2015. – 19 с. (Законы РФ)
3. Семейный кодекс Российской Федерации. – М.: Эксмо, 2012. – 96 с. – (Актуальное законодательство)
4. Об образовании в Российской Федерации: Федер. закон: [принят Гос. Думой 21 декабря 2012 г., одобрен Советом Федерации 26 декабря 2012 г.]. – М.: Эксмо, 2014. – 208 с. – (Актуальное законодательство)
5. Концепция развития дополнительного образования детей до 2030 г.: [распоряжение Правительством Российской Федерации 31 марта 2022 г. № 678-р]. [Электронный ресурс] // Режим доступа: <http://static.government.ru/media/files/3flgkklAJ2ENBbCFVEkA3cTOsiypicBo.pdf>
6. "Об утверждении санитарных правил СП 2.4. 3648-20 "Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи". [Постановление Главного государственного санитарного врача Российской Федерации от 28.09.2020 № 28] Официальный интернет-портал правовой информации. [Электронный ресурс] // Режим доступа: <http://publication.pravo.gov.ru/Document/View/0001202012210122>
7. Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным образовательным программам: [Приказ Министерства образования и науки Российской Федерации от 27.07.2022 № 629.] Официальный интернет-портал правовой информации. [Электронный ресурс] // Режим доступа: <http://publication.pravo.gov.ru/Document/View/0001202209270013>

Список литературы для педагога

1. Васильев, А. Н. Python на примерах [Текст]: практ. курс / А. Н. Васильев - Наука и Техника, 2019 - 432 с.
2. Прохоренок, Н. А. Python 3: самое необходимое [Текст]: практ. курс / Н. А. Прохоренок, В. А. Дронов - БХВ-Петербург, 2019 - 608 с.
3. Гэддис, Т. Начинаем программировать на Python [Текст]: учебник/Т. Гэддис - БХВ-Петербург, 2019 - 768 с.
4. Седжвик, Р. Программирование на языке Python /Р. Седжвик, К. Уэйн, Р. Дондеро - Вильямс, 2017 - 736 с.
5. Харрисон, М. Как устроен Python. [Текст]: практ. курс / М. Харрисон - Питер, 2002 - 272 с.

Список литературы для учащихся

Не предусмотрен

Интернет ресурсы

<https://www.stepik.ru/>

<https://wingware.com/>

<https://python.org/>

<https://code.visualstudio.com/>