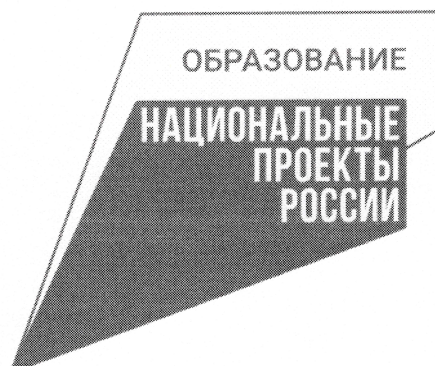


Муниципальное бюджетное общеобразовательное
учреждение «Гимназия №21»
Центр цифрового образования детей «IT-cube.Чита»



Дополнительная общеобразовательная

(общеразвивающая) программа

«Основы программирования (3-5 класс)»

Категория учащихся: учащиеся 3-5 классов средней школы

Срок освоения программы: 72 часа.

Направление: Основы программирования

Составитель программы:
Рожественская Анастасия Владимировна

Утверждена приказом Директора
№ 40 от 25 августа 2022 г.



Чита, 2022

Оглавление

1. Пояснительная записка	2
2. Новизна данной образовательной программы	11
3. Общая характеристика курса «Основы программирования»	12
3.1. Основные разделы программы	12
3.2. Формы организации учебных занятий	19
4. Личностные, метапредметные результаты освоения курса «Основы программирования»	23
5. Перечень учебно-методического обеспечения	26
6. Учебно-тематический план	28
7. Список литературы	29

1. Пояснительная записка

1.1. Направленность образовательной программы

Направленность настоящей общеобразовательной программы – научно-техническая. Программа является прикладной, носит практико-ориентировочный характер и направлена на овладение обучающимися представлений о возможностях сред программирования, приобретение ими опыта работы в команде, освоения навыков программирования, знакомства с основными понятиями, принципами и инструментариями разработки программирования на языке Python, формирование аналитического, творческого и критического мышления, теоретических и практических навыков. Реализация образовательной программы способствует интеллектуальному развитию обучающихся, раскрытию их талантов. Содействует профессиональному самоопределению обучающихся, развитию их познавательной активности и творческой самореализации.

1.2. Актуальность и практическая значимость образовательной программы

Изучение основ программирования на языке Python, с которыми познакомятся обучающиеся в рамках курса, станет продолжением их познаний в программировании. Через изучение основ программирования на языке Python будут развиваться исследовательские, инженерные и проектные компетенции.

Ключевые навыки, приобретаемые в ходе освоения образовательной программы, компетенции HardSkills и SoftSkills, подразумевает получение ряда базовых компетенций.

Педагогическая целесообразность программы заключается в том, что дети приобретут практические навыки, которые станут основой для дальнейшего изучения основ программирования. Методы, применяемые в процессе обучения, такие как проблемное обучение, проектная деятельность, способствуют формированию мотивации

обучающихся к углубленному изучению программирования, как одной из компьютерных наук. У детей формируется познавательный интерес, самостоятельность мышления, стремление к самопознанию.

Осваивая данную программу, обучающиеся будут овладевать навыками, которые будут востребованы в ближайшие десятилетия в специальностях, многие из которых включены в Атлас профессий будущего. Практически для каждой перспективной профессии будут полезны знания и навыки, получаемые в процессе обучения по программе.

Отличительная особенность программы состоит в том, что она позволяет привлечь детей среднего школьного возраста к изучению программирования при помощи языка Python, так как он обладает следующими достоинствами:

- Python – это текстовый язык программирования. Он универсален, пригоден для создания самых разных программ, от текстовых процессоров до веб-браузеров;
- Python – простой и удобный язык. По сравнению со многими другими языками читать и составлять программы на Python совсем не сложно;
- В Python есть библиотеки готовых процедур для использования в своих программах. Это позволяет создавать сложные программы быстро;
- Python используется как язык программирования крупными корпорациями, такими как Google.

Адресат программы - дети от 9 до 12 лет. Наполняемость групп: 10 – 15 человек.

Объем и срок освоения программы. Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» (стартовый уровень) рассчитана на 1 год обучения.

Дети, освоившие стартовый уровень, могут продолжить обучение на базовом уровне программы «Программирование на языке Python» (базовый уровень) без входного контроля.

Режим занятий: 72 академических часа в год, 2 академических часа в неделю.

В каникулярное время занятия проводятся в соответствии с календарным учебным графиком, допускается изменение форм занятий, проведение воспитательных мероприятий.

1.3 Способы реализации образовательной программы

Для повышения эффективности освоения обучающимися настоящей образовательной программы целесообразным является разделение процесса обучения на несколько логически связанных ключевых этапов (модулей):

- приобретение обучающимися теоретических сведений об основах программирования на языке Python;
- освоение базовых принципов работы в среде Pycharm;
- приобретение обучающимися теоретических сведений и практических навыков программирования на языке Python (линейный алгоритм, условные операторы, циклы, списки, кортежи, словари);
- генерация идей и их реализация их;
- включение в творческую деятельность.

По итогам освоения образовательной программы обучающийся сможет самостоятельно написать программу на языке Python, запустить ее, оценить результаты.

Реализация образовательной программы происходит преимущественно индивидуально. При реализации образовательной

программы в ходе планирования и проведения занятий учитываются индивидуальные особенности обучающихся. Это, в свою очередь, позволяет заинтересовать, вовлечь в образовательный процесс всех обучающихся, поддерживать высокий уровень мотивации к образовательной деятельности каждого из них на протяжении всего периода реализации образовательной программы.

В ходе реализации образовательной программы рекомендуется использовать следующие формы занятий:

- на этапе изучения нового материала – лекция, объяснение, рассказ, демонстрация;
- на этапе закрепления изученного материала – беседа, дискуссия, практическая работа, дидактическая или педагогическая игра;
- на этапе повторения изученного материала – наблюдение, устный контроль (опрос, игра), творческое задание;
- на этапе проверки полученных знаний – выполнение дополнительных заданий, публичное выступление с демонстрацией результатов работы.

Реализация образовательной программы предусматривает проведения занятий следующих форм: беседа, круглый стол, демонстрации, объяснение, практическая работа на компьютере, самостоятельная работа, ролевые и деловые игры, проектная деятельность, консультация, групповые работы.

При реализации образовательной программы применяются активные методы обучения такие, как:

- модульный метод (методика проблемного обучения).

1.4 Цель образовательной программы

Целью данной общеобразовательной программы является формирование знаний, умений и навыков программирования на языке

Python, развитие интереса обучающихся к информационным технологиям, реализация их творческих идей в области программирования в виде решения задач различного уровня сложности.

1.5. Задачи образовательной программы

Обучающие:

- сформировать у детей представление об основных элементах программирования;
- познакомить с синтаксисом языка программирования Python;
- сформировать у детей навыки работы в интегрированной среде разработки на языке Python;
- способствовать приобретению навыков разработки эффективных алгоритмов и программ на основе изучения языка программирования Python у детей.

Развивающие:

- совершенствовать аналитические навыки;
- формировать творческий подход к поставленной задаче;
- формировать навык алгоритмического и логического мышления;
- совершенствовать навык поиска информации в сети Интернет, анализа выбранной информации на соответствие запросу, использование информации при решении задач;
- развивать умение планировать свои действия с учётом фактора времени;

- способствовать развитию творческих способностей учащихся, познавательных интересов, развитию индивидуальности и самореализации;
- развивать пространственное воображение, внимательность к деталям, ассоциативное и аналитическое мышление;
- развивать познавательные способности обучающегося, память, внимание, пространственное мышление, аккуратность и изобретательность при выполнении учебных проектов;
- развивать навыки инженерного, системно-комбинаторного мышления и эффективного использования электронного вычислительного оборудования.

Воспитательные:

- воспитывать в детях усидчивость, аккуратность, умение доводить начатое дело до конца;
- формировать коммуникативные навыки.
- развить у обучающихся чувство внутренней инициативы, самостоятельности;
- привить обучающемуся тягу к самосовершенствованию;
- воспитать мотивацию учащихся к изобретательству, созданию собственных проектов, ответственности за результат своей работы на компьютере и за возможные ошибки;
- привить стремление к получению качественного законченного результата в проектной деятельности;
- воспитывать социально-значимые качества личности человека: ответственность, коммуникабельность, добросовестность, взаимопомощь, доброжелательность, критичность и самокритичность мышления.

Мотивирующие:

- мотивировать обучающихся к нестандартному мышлению, изобретательству и инициативности при выполнении задач;
- поощрять у учащихся мотивацию к работе в формате «от идеи до законченного проекта» на всех этапах деятельности;
- поощрять инициативу обучающихся предлагающих нестандартное решение задач и их реализацию;
- поддерживать стремление к самостоятельному повышению уровня навыков программирования, моделирования и визуализации, необходимых для поддержания конкурентоспособности специалиста в современном высокотехнологичном мире.

1.6 Категория обучающихся

Образовательная программа рассчитана на обучающихся в возрасте от 9 до 12 лет (3 – 5 класс). Набор в группы – свободный (без предъявления дополнительных требований к уровню входных компетенций). Рекомендуемое количество обучающихся в группе от 10 до 15 человек.

1.7 Планируемые (прогнозируемые) результаты обучения

К концу реализации программы обучающиеся будут знать:

- основные классические алгоритмы и способы их реализации;
- синтаксис языка программирования Python;
- простые и сложные структуры данных, а также конструкции для работы с ними;
- некоторые модули (turtle, random, tkinter и др.).

- основные элементы программирования: ввод (вывод) информации в память (из памяти), данные, операции с данными, условное выполнение, циклы, подпрограммы
- К концу реализации программы обучающиеся будут уметь:
- разбивать решение задачи на подзадачи, то есть составлять алгоритм;
- объяснять и использовать на практике как простые, так и сложные структуры данных и конструкции для работы с ними;
- искать и обрабатывать ошибки в коде;
- писать грамотный, красивый код;
- анализировать как свой, так и чужой код;
- импортировать модули в программу;
- работать с информацией: находить, оценивать и использовать информацию из различных источников, необходимую для решения профессиональных задач (в том числе на основе системного подхода);
- грамотно строить коммуникацию, исходя из целей и ситуации;
- работать с вычислительной техникой.

К концу реализации программы обучающиеся будут иметь представление:

- о возможностях языка программирования Python;
- о структурном программировании.

Полученные в ходе реализации образовательного курса знания, умения и навыки могут быть применены в ходе реализации последующих образовательных программ.

2. Новизна данной образовательной программы

Новизна программы состоит в том, что она построена таким образом, чтобы помочь обучающимся развить свои навыки в программировании, или получить их, используя инструменты языка Python.

Данная программа может стать толчком к раскрытию личности и творческого потенциала ребенка, т.к. не загоняет его в конкретные рамки, а соревновательный момент будет способствовать самореализации обучающегося и поможет ему лучше адаптироваться в современном мире

Особенности заключается во внедрении в образовательный процесс новых технологий, благодаря которому происходит вовлечение детей в техническую и конструкторскую деятельность. Кроме этого, в программе реализуется творческий подход ребёнка к продукту своей деятельности, что способствует развитию личности и способностей к техническому творчеству.

Новизна программы состоит в том, что она построена таким образом, чтобы помочь обучающимся развить свои навыки в программировании, или получить их, используя инструменты языка Python.

Данная программа может стать толчком к раскрытию личности и творческого потенциала ребенка, т.к. не загоняет его в конкретные рамки, а соревновательный момент будет способствовать самореализации обучающегося и поможет ему лучше адаптироваться в современном мире

3. Общая характеристика курса «Основы программирования»

3.1. Основные разделы программы

Раздел 1. Введение в язык программирования Python

Тема 1.1. Устройство языка Python. Среда разработки IDLE. Сохранение Python-программ

Теория. Техника безопасности на занятии. Понятие «алгоритм», «исполнитель», «язык программирования», «программа», «интерпретатор». История языка программирования Python и его возможности. Виды окон в IDLE: окно программы и окно консоли. Сравнение этих окон и их возможностей.

Практика. Сохранение и запуск python-программ в среде разработки IDLE.

Тема 1.2. Вывод данных на языке Python. Ввод и отладка программ в среде

Теория. Правила синтаксиса Python: правило начала, правило порядка, правило регистра. Понятие функции. Функция print().

Практика. Проект «Символьная графика». Создание определённого рисунка с помощью символов. Отработка функции print().

Тема 1.3. Переменные

Теория. Понятие «переменная». Правила именования переменных в языке Python. Оператор присваивания. Функция input().

Практика. Проект «Аватар». В данном проекте отрабатывается функция input(), с помощью которой становится возможным ввести свои данные в программу и отобразить их.

Тема 1.4. Выражения. Типы данных

Теория. Арифметические операции с помощью математических

операторов +, -, *, /. Порядок выполнения операций. Понятие «выражение», «типы данных». Функции int() и str().

Практика. Проект «Сумматор». При написании данной программы отрабатываются математические операторы и функции int() и str().

Тема 1.5. Строки и списки

Теория. Понятие «строка». Создание строк. Переменные внутри строк. Операции со строками. Понятие «список». Создание списков. Добавление/удаление элементов в/из список/списка. Операции со списками.

Практика. Решение задач на отработку операций со строками и списками.

Тема 1.6. Кортежи и словари

Теория. Понятие «кортеж». Создание кортежа. Операции с кортежем. Понятие «словарь». Создание словаря.

Практика. Проект «Любимые вещи»: создание списка любимых развлечений и любимых лакомств.

Тема 1.7. Рисование с помощью «Черепашьей графики»

Теория. Понятие «модуль». Модуль turtle. Импорт модуля. Создание холста. Перемещение черепашки.

Практика. Проект «Я рисую»: рисование изображения, состоящего из линий с помощью команд модуля на холсте.

Раздел 2. Основы языка Python

Тема 2.1. Условный оператор

Теория. Понятие «условный оператор», «вложенные команды», «оператор сравнения». Конструкция if и её синтаксис. Операторы сравнения: <, >, >=, <=, !=, ==. Структура программы. Конструкция if-

else. Команды if и elif.

Практика. Решение задач на отработку условного оператора и операторов сравнения.

Тема 2.2. Объединение условий

Теория. Логические операторы: and, or, not. Порядок выполнения операций. Переменные без значения – None.

Практика. Проект «Калькулятор»: создание приложения по определенным условиям.

Тема 2.3. Использование цикла for

Теория. Понятие «цикл», «цикл со счётчиком». Конструкция for и её синтаксис.

Практика. Проект «Таблица умножения»: создание приложения по определенным условиям.

Тема 2.4. Использование цикла while

Теория. Понятие «цикл с предусловием». Конструкция while и её синтаксис. Зацикливание и выход из цикла с помощью команды break.

Практика. Проект «Бомба взорвалась!». Написание программы по определенным условиям.

Тема 2.5. Решение задач на циклические алгоритмы

Теория. Виды циклов и их конструкции.

Практика. Решение задач на применение циклов for и while.

Тема 2.6. Применение функций

Теория. Понятие «функция», «параметр функции», «значение функции». Строение функции: имя, аргумент, тело. Создание и вызов функции. Переменные и область видимости.

Практика. Решение задач на отработку понятия «функция», её строение и синтаксис.

Тема 2.7. Использование встроенных функций

Теория. Функции: abs, bool, dir, eval, exec, float, int, len, max, min, range, sum.

Практика. Решение задач на отработку функций.

Тема 2.8. Применение модулей

Теория. Понятие «модуль». Импортирование модуля в программу. Полезные модули: random, time, pickle.

Практика. Решение задач на применение модулей.

Тема 2.9. Классы и объекты

Теория. Понятие «объект». Концепция объектов. Понятие «класс». Использование классов в Python. Инициализация объектов.

Практика. Решение задач на отработку умения создавать классы и объекты.

Тема 2.10. События

Теория. Понятие «событие», «подписка на событие», «обработка события».

Практика. Проект «Поймай черепашку». Пишется приложение-игра, в котором создается холст и на нём прорисовывается черепашка, при нажатии на которую происходит ее перемещение в новое место на холсте.

Тема 2.11. Работа с файлами

Теория. Понятие «файл». Классификация файлов в зависимости от видов информации. Создание текстового файла. Открытие файла в Python.

Запись в файл.

Практика. Проект «Блокнот»: создание простейшего текстового редактора в виде приложения.

Тема 2.12. «Черепашья графика»

Теория. Рисование фигур с использованием управляющих конструкций.

Практика. Проект «Фантастический мир»: рисование фигур, параметры которых задаются пользователем при запуске программы.

Тема 2.13. Графика с модулем tkinter

Теория. Модуль tkinter и его возможности.

Практика. Проект «Анимация»: отрисовывание объекта на холсте и программирование анимации.

Раздел 3. Игры на Python

Тема 3.1. Игра «Прыг-скок!»

Теория. Обсуждение сюжета игры: объекты и их роль. Создание игрового холста. Создание главного цикла игры. Отрисовка мяча. Программирование мяча: перемещение, отскоки от границ игрового холста, первоначальное направление движения мяча. Отрисовка ракетки. Программирование ракетки: управление, установления границ. Обработка события столкновения мяча и ракетки. Обработка события «проигрыш». Доработка игры: задержка перед началом игры; экран «Конец игры»; ускорение мяча; счет в игре.

Практика. Приложение-игра «Прыг-скок!»: неуправляемый пользователем объект «мяч» и управляемые пользователем объект «ракетка».

Тема 3.2. Игра «Охотник за пузырями»

Теория. Обсуждение сюжета игры: объекты и их роль. Создание

игрового холста. Создание главного цикла игры. Отрисовка подложки. Программирование подложки: перемещение, поворот, торможение при касании границы холста. Отрисовка пузырей разного размера и их различное положение на холсте с помощью модуля random. Программирование пузырей: движение. Обработка события касания объектов «подводная лодка» и «пузырь». Доработка игры: задержка перед началом игры; счет в игре и его отображение; создание лимита времени; экран «Конец игры».

Практика. Приложение-игра «Охотник за пузырями»: управляемый пользователем объект «подводная лодка» и неуправляемый объект «пузырь», создаваемые в рандомном количестве. Задача пользователя: управлять лодкой и протыкать пузыри, зарабатывая при этом баллы.

Тема 3.3. Игра «Человечек спешит к выходу»

Теория. Обсуждение сюжета игры: объекты и их роль. Создание игрового холста. Создание главного цикла игры. Отрисовка человечка (три изображения для анимации: влево, вправо, на месте). Программирование человечка: анимация, управление. Отрисовка платформ трёх видов: маленькой длины, средней и большой. Размещение платформ на холсте игры. Отрисовка двери: закрытая и открытая. Обработка событий столкновений человечка с объектами: «дверь» и «платформа». Доработка игры: экран «Вы победили!»; экран «Конец игры»; создание резерва жизней для игрока; создание второго уровня с движущимися платформами; создание третьего уровня с противниками на движущихся платформах.

Практика. Приложение-игра «Человечек спешит к выходу». Сюжет игры: Человечек очутился в плену у злодея, и ваша задача – помочь человечку спастись, добравшись до выхода на верхнем этаже. Человечек выглядит как фигурка, которая может двигаться вправо, влево, а также прыгать. На этажах расположены платформы, на которые человечку предстоит запрыгивать. Цель игры – добраться до двери выхода.

Раздел 4. Итоговый проект

Практика. Разработка собственного приложения-игры с сюжетом. Условия игры: обязательно должен присутствовать управляемый пользователем персонаж, должна быть прорисована и запрограммирована игровая обстановка.

3.2. Формы организации учебных занятий

Формы организации образовательного процесса:

- фронтальная – подача материала всей учебной группе обучающихся;
- индивидуальная – самостоятельная работа обучающихся с оказанием педагогом помощи при возникновении затруднения;
- групповая – предоставление учащимся возможности самостоятельно построить свою деятельность, ощутить помощь со стороны друг друга, учесть возможности каждого на конкретном этапе деятельности.

Формы организации учебных занятий:

- комбинированный тип (аудиторные занятия) проведения обязательных занятий, сочетающий в себе элементы теории и практики;
- самостоятельная (внеаудиторная) работа, в которой используются разные по уровню сложности задания и носят репродуктивный и творческий характер.
- контрольная (проверочная) работа (аудиторные занятия) для контроля освоения знаний, умений.
- резервные часы используются преподавателем как запасные занятия для повторения и углубленного изучения тем, более подробного разбора решения задач, при подготовке к контрольным работам, проведения работы над ошибками и т.д.

На каждом занятии присутствуют теоретическая и практическая части, проводится повторение и закрепление умений и знаний, полученных на предыдущем занятии, проводится проверка выполненного домашнего задания. Основные задания являются обязательными для выполнения всеми обучающимися в группе,

выполняются на компьютере с использованием интегрированной среды разработки.

В ходе обучения проводится промежуточное тестирование по темам для определения уровня знаний обучающихся. ДОП «Программирование на Python» заканчивается написанием программы для решения одной из задач.

Методы организации учебного процесса:

- вводное занятие – педагог знакомит обучающихся с техникой безопасности, особенностями организации деятельности и предлагаемым планом работы на текущий год;
- ознакомительное занятие – педагог знакомит обучающихся с новыми методами работы в зависимости от темы занятия;
- тематическое занятие – на котором детям предлагается работать над моделированием по определенной теме. Занятие содействует развитию творческого воображения обучающихся;
- занятие-проект – на таком занятии обучающиеся получают полную свободу в выборе направления работы, не ограниченного определенной тематикой. Обучающиеся, участвующие в работе по выполнению предложенного задания, рассказывают о выполненной работе, о ходе выполнения задания, о назначении выполненного проекта;
- конкурсное игровое занятие – строится в виде соревнования для повышения активности обучающихся и их коммуникации между собой;
- комбинированное занятие – проводится для решения нескольких учебных задач;

- итоговое занятие – служит подведению итогов работы за учебный год, может проходить в виде мини-выставок, просмотров творческих работ и презентаций.

Формы и методы контроля:

Педагогический мониторинг позволяет систематически отслеживать результативность реализации программы. Мониторинг включает в себя традиционные формы контроля: промежуточную и итоговую аттестацию результатов обучения детей.

Промежуточная аттестация проводится в конце первого полугодия.

Итоговый контроль проводится в конце года с целью определения степени достижения результатов обучения и получения сведений для совершенствования программы и методов обучения.

Аттестация обучающихся может проходить на итоговом занятии в форме презентации своего проекта.

Итоговая аттестация учащихся осуществляется по 100-балльной шкале, которая переводится в один из уровней освоения образовательной программы согласно таблице:

Баллы, набранные учащимся Уровень освоения	Баллы, набранные учащимся Уровень освоения
0–50 баллов Низкий	0–50 баллов Низкий
50–75 баллов Средний	50–75 баллов Средний
75–100 баллов Высокий	75–100 баллов Высокий

Общая характеристика учебного процесса:

Алгоритм учебного занятия:

1. Мотивация обучающихся.

2. Актуализация имеющихся знаний.
3. Теоретический блок нового материала.
4. Закрепление материала.
5. Перерыв.
6. Теоретический блок нового материала.
7. Закрепление материала.
8. Рефлексия.

4. Личностные, метапредметные результаты освоения курса «Основы программирования»

Предметные результаты:

- умение определять результат выполнения алгоритма при заданных
- исходных данных, узнавать изученные алгоритмы обработки чисел и числовых последовательностей, создавать на их основе несложные программы анализа данных, читать и понимать несложные программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня;
- понимание основных предметных понятий («информация», «алгоритм», «исполнитель», «модель») и их свойств;
- развитие логических способностей и алгоритмического мышления, умения составить и записать алгоритм для конкретного исполнителя, знакомство с основными алгоритмическими структурами – линейной, условной и циклической;
- умение выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных;
- навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ; умение использовать основные управляющие конструкции объектно-ориентированного программирования и библиотеки прикладных программ, выполнять созданные программы;
- умение разрабатывать и использовать компьютерно-математические модели, оценивать числовые параметры моделируемых объектов и процессов, интерпретировать результаты, получаемые в ходе моделирования реальных процессов, анализировать готовые модели на предмет соответствия реальному объекту или процессу;
- умение соблюдать нормы информационной этики и права.

Личностные результаты:

- формирование ответственного отношения к учению, способности довести до конца начатое дело аналогично завершённым творческим учебным проектам;
- формирование способности к саморазвитию и самообразованию
- средствами информационных технологий на основе приобретённой благодаря иллюстрированной среде программирования мотивации к обучению и познанию;
- развитие опыта участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками в процессе образовательной, учебно- исследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня;
- формирование целостного мировоззрения, соответствующего
- современному уровню развития информационных технологий;
- формирование осознанного позитивного отношения к другому человеку, его мнению, результату его деятельности;
- формирование ценности здорового и безопасного образа жизни; усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

Метапредметные результаты:

- умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
- умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата; понимание, что в программировании длинная программа не всегда лучшая;
- умение критически оценивать правильность решения учебно-исследовательской задачи;
- умение корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;

- владение основами самоконтроля, способность к принятию решений;
- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий (ИКТ-компетенция);
- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно-исследовательской деятельности.

5. Перечень учебно-методического обеспечения

Образовательный процесс осуществляется в очной форме.

В образовательном процессе используются следующие методы:

1. словесные (беседа, опрос, дискуссия и.т.д.);
2. игровые;
3. метод проблемного изложения (постановка проблемы и решение её самостоятельно или группой);
4. метод проектов;
5. наглядные:
 - a. демонстрация плакатов, схем, таблиц, диаграмм;
 - b. использование технических средств;
 - c. просмотр кино- и телепрограмм;
6. практические:
 - a. практические задания;
 - b. анализ и решение проблемных ситуаций и т.д.

Методическое обеспечение

Методические пособия, разработанные преподавателем с учётом конкретных задач, варианты демонстрационных программ, материалы по терминологии ПО, инструкции по настройке оборудования, учебная и техническая литература. Используются педагогические технологии индивидуализации обучения и коллективной деятельности.

Методические учебные пособия:

1. Бриггс, Джейсон, Python для детей. Самоучитель по программированию / Джейсон Бриггс ; пер. с англ.

Станислава Ломакина ; [науч. ред. Д. Абрамова]. — М. : Манн, Иванов и Фербер, 2017. — 320 с.

2. Великович Л., Цветкова М. Программирование для начинающих. – М.:Бином, 2008.
3. Программирование для детей. Перевод с английского Станислава Ломакина, Москва, «Манн, Иванов и Фербер», 2015 г.

Оборудование:

1. Компьютерный класс (15 ученических ПК + 1 учительский ПК) с выходом в сеть Интернет
2. Маркерная доска
3. Проектор
4. Принтер

Сетевые образовательные ресурсы:

1. Сайт pythonworld.ru – «Python 3 для начинающих».
2. Сайт pythontutor.ru – «Питонтьютор».
3. <https://www.youtube.com/playlist?list=PLJOzdkh8T5kpIBTG9mM2wVBjh-5OpdwBI> – Лекции А.В. Умнова, прочитанные в Школе анализа данных Яндекса.

6. Учебно-тематический план

72 академических часа в год

№ п/п	Название разделов, тем	Количество часов		
		Всего	Теория	Практика
1	Введение в язык программирования Python	16	8	8
1.1	Устройство языка Python. Среда разработки IDLE. Сохранение Python-программ	2	1	1
1.2	Вывод данных на языке Python. Ввод и отладка программ в среде	2	1	1
1.3	Переменные	2	1	1
1.4	Выражения. Типы данных	2	1	1
1.5	Строки и списки	2	1	1
1.6	Кортежи и словари	2	1	1
1.7	Рисование с помощью «Черепашьей графики»	4	2	2
2	Основы языка Python	32	16	16
2.1	Условный оператор	2	1	1
2.2	Объединение условий	2	1	1
2.3	Использование цикла for	2	1	1
2.4	Использование цикла while	2	1	1
2.5	Решение задач на циклические алгоритмы	2	1	1
2.6	Применение функций	2	1	1
2.7	Использование встроенных функций	2	1	1
2.8	Применение модулей	2	1	1
2.9	Классы и объекты	2	1	1
2.10	События	2	1	1
2.11	Работа с файлами	2	1	1
2.12	«Черепашья графика»	4	2	2
2.13	Графика с модулем tkinter	6	3	3
3	Игры на Python	20	4	16
3.1	Игра «Прыг-скок!»	6	1	5
3.2	Игра «Охотник за пузырями»	6	1	5
3.3	Игра «Человечек спешит к выходу»	8	2	6
4	Итоговый проект	4	-	4
	Итого:	72	28	44

7. Список литературы

Литература, использованная при подготовке программы:

1. Бизли Д.М., Г. Ван Россум. Язык программирования Python. Справочник. (пер. с англ.) Киев: ДиаСофт, 2000.
2. Информатика: Учебник для 10-11 класса / Н.Д. Угринович - М.: Бином, 2009. - 512 с.
3. Лутц М. Программирование на Python. (пер. с англ.) СПб.:Символ- Плюс., 2002.
4. Россум, Г. и др. Язык программирования Python. 2001. 454 с.
5. Сузи Р.А. Python. Наиболее полное руководство. СПб.: БХВ-Петнбург., 2002.
6. Сузи Р.А. Язык программирования Python. М: Бином. Лаборатория знаний. - 2006.
7. Фридланд А.Я., Ханамирова Л.С., Фридланд И.А. Информатика и компьютерные технологии. Основные термины. Толковый словарь. М..ИздательствоАстрель., 2003.
8. Хахаев И.А. Практикум по алгоритмизации и программированию на Python: / И.А. Хахаев - М.: Альт Линукс, 2010. - 126 с.
9. М. Лутц. Изучаем Python. СПб.: Символ-Плюс, 2011.
10. Задачи по программированию. Под ред. С. М. Окулова, М.: БИНОМ. Лаборатория знаний, 2006.
11. С. М. Окулов. Основы программирования. М.: Бином. Лаборатория знаний, 2012.
12. М. Лутц. Изучаем Python. СПб.: Символ-Плюс, 2011.

13. Информатика и ИКТ. Задачник-практикум в 2 частях.
Под ред. И. Г. Семакина и Е. К. Хеннера. М.: БИНОМ.
Лаборатория знаний, 2014. Revision: 226.

Литература, рекомендованная обучающимся:

1. Вордерман Кэрол. Программирование для детей. 2016. - 357с.
2. Джейсон Биггс. Python для детей. 2017. – 319с.
3. ПейнБрайсон. Python для детей и родителей. 2017. – 352с.

Ресурсы в интернете

4. <https://pythonworld.ru/samouchitel-python>
5. <http://server.aesc.msu.ru/materials/PYTHON/pythonworldru.pdf>