

Муниципальное общеобразовательное учреждение
«Средняя общеобразовательная школа №3 г. Ершова Саратовской области»

ПРИНЯТА на заседании педагогического совета МОУ «Средняя общеобразовательная Школа №3 г. Ершова Саратовской области» Протокол № 13 от 19.04.2023	УТВЕРЖЕНО директор МОУ «Средняя общеобразовательная Школа №3 г. Ершова Саратовской области» Приказ № 140 от 24.04.2023  А.В. Широкова
--	---

Дополнительная общеобразовательная общеразвивающая
программа

технической направленности
«Программирование на Python»
(стартовый уровень)

Направленность: техническая
Форма реализации: очная
Возраст обучающихся: 10 - 12 лет
Срок реализации: 1 год

Автор – составитель:
Бурова Ольга
Валерьевна –
педагог
дополнительного
образования

Раздел №1 «Комплекс основных характеристик программы»

1.1. Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа «Программирование на Python» (стартовый уровень) муниципального общеобразовательного учреждения «Средняя общеобразовательная школа № 3 г. Ершова Саратовской области» разработана в рамках **технической направленности** в соответствии с:

- «Закон об образовании в Российской Федерации» (№ 273-ФЗ от 29 декабря 2012 г.);
- «Порядок организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам» (пр. Министерства просвещения РФ от 27 июля 2022 г. №629)
- Приказ Министерства образования и науки Российской Федерации от 23 августа 2017 года №816 «Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;
- Постановление Главного государственного санитарного врача РФ от 28 сентября 2020 года №28 «Об утверждении СанПиН 2.4.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»
- Положением о дополнительной общеобразовательной общеразвивающей программе МОУ «СОШ № 3 г. Ершова Саратовской области»

Актуальность программы. В современную жизнь человека все шире внедряются компьютеры и информационные технологии. Поэтому все большее значение приобретает компьютерная грамотность.

Курс изучения компьютерной грамотности состоит из двух разделов: пользовательского курса и программирования. Раздел «Программирование» в школьном курсе представлен языком программирования Pascal, а многим учащимся хочется познакомиться с другими языками программирования, самим попробовать разработать программы, которые можно использовать на уроках и во внеурочной деятельности. Данная Программа позволяет реализовать эти желания, так как уделяется большое внимание практической работе учащихся на компьютере, самостоятельной разработке ими программ для решения практических задач.

Новизна. Python дает более широкие возможности в области программирования, чем Pascal, который входит в школьный курс информатики. На языке Python можно легко и быстро создавать простые компьютерные игры, трёхмерные модели и программировать роботов. Этот язык быстрее и легче усваивается, чем Pascal. Многие мировые компании такие, как Intel, Cisco, Hewlett- Packard, используют этот язык при реализации своих проектов. Крупнейшие интернет-ресурсы такие, как Google, YouTube, также разработаны с помощью языка программирования Python.

Педагогическая целесообразность программы «Программирование на PYTHON (стартовый уровень)» заключается в привлечении учащихся к занятиям техническим творчеством, что способствует развитию логического мышления, творческих способностей и навыков решения задач программирования. Программирование мотивирует к занятиям в различных научных областях (физики, информатики, алгебры, геометрии и др.), развивает воображение и способствует ранней профориентации подростков. Для достижения поставленных задач занятия проводятся в формате «от простого к сложному». Учащиеся вспоминают свои знания по основам алгоритмизации и программирования и на их основе, углубляя их, учатся составлять простые и сложные программы.

Отличительные особенности программы. Данная Программа разработана на

основании программы «Программирование на языке Python» (разработчик Киселева Н.Н., педагог дополнительного образования ГБОУ Школа № 1375). В отличие от базовой в данную программу внесены следующие изменения:

- иначе сформулирована цель программы;
- выделены группы предполагаемых результатов: личностные, метапредметные, предметные;
- добавлены возрастные особенности обучающихся;

Основное количество часов отводится практическому написанию программ. Каждый обучающийся реализует индивидуальный проект в результате освоения программы. Продукт, полученный в результате освоения программы, имеет прикладной характер и может быть использован по необходимости.

Адресат программы, возрастные особенности. Возраст детей, участвующих в реализации программы 10-12 лет, программа построена с учетом возрастных и индивидуальных особенностей учащихся, их потенциальных возможностей, и рассчитана на творческую, познавательно-интеллектуальную, самостоятельную деятельность учащихся.

Возрастные особенности обучающихся. В возрасте 9 - 11 лет ребенок начинает быть самостоятельным. Ведущая деятельность – учебная. Это время освоения знаний и развития интеллектуально-познавательной деятельности. Одно из новообразований – формирование внутреннего плана действий, имеет общие моменты с алгоритмизацией и программированием, поэтому изучение робототехники вполне оправдано. Наиболее эффективные средства обучения – готовые роботы и конструкторы. Именно в этом возрасте развиты фантазия и воображение, это позволяет детям создавать свои уникальные фигуры, накапливается жизненный и интеллектуальный опыт, способствующий развитию детей и помогающий от образного мышления перейти к логическому виду мышления.

Наполняемость группы 12 человек.

Объем и срок реализации программы: 1 год, 144 часа.

Режим занятий: Занятия проводятся 2 раза в неделю по 2 часа продолжительностью 45 минут с перерывом между занятиями 15 минут.

Так как в течение учебного года возникает непреодолимая сила, или форс-мажор – обстоятельства (эпидемия, карантин, погодные условия и прочее), не позволяющие осуществлять обучение в обычной (очной) форме, реализация программы возможна с помощью электронных (дистанционных) технологий.

1.2. Цель и задачи программы

Цель: развитие инженерного мышления и воспитание конкурентно способной личности через обучение учащихся программированию посредством языка Python.

Задачи:

Обучающие:

- обучить языку программирования Python и созданию программ на его основе;
- научить создавать прикладное программное обеспечение;
- расширять кругозор обучающихся в области программирования;
- научить дизайнерскому оформлению созданного программного обеспечения.

Развивающие:

- сформировать трудовые умения и навыки, умение планировать свои действия с учётом фактора времени, в обстановке с элементами конкуренции, предвидеть результат и достигать его, при необходимости вносить коррективы в первоначальный замысел;
- способствовать развитию логического мышления и технических навыков;
- способствовать развитию навыков исследовательской и проектной деятельности;

- сформировать и развить навыки работы с различными источниками информации, умения самостоятельно искать, извлекать и отбирать необходимую для решения учебных задач информацию;
- познакомить с правилами индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

Воспитательные:

- способствовать воспитанию этики групповой работы;
- способствовать развитию основы коммуникативных отношений внутри проектных групп и в коллективе в целом;
- способствовать воспитанию упорства в достижении результата;
- сформировать целеустремленность, организованность, ответственное отношение к труду и уважительное отношение к окружающим.

1.3. Планируемые результаты

Метапредметные результаты

По окончании программы обучающийся:

- научится самостоятельно ставить и формулировать для себя новые задачи, планировать свои действия, планировать пути решения поставленной задачи для получения эффективного результата, корректировать свои действия;
- получит технические навыки и развитое логическое мышление;
- получит навыки исследовательской и проектной деятельности;
- научится работать с различными источниками информации, извлекать нужную информацию из открытых источников;
- усвоит правила индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

Личностные результаты

По окончании программы обучающийся:

- овладеет способностью к саморазвитию и самообразованию, будут развиты любознательность, внимательность и настойчивость при выполнении заданий практического характера;
- научится организовывать учебное сотрудничество и совместную деятельность с педагогом и сверстниками в процессе проектной и учебно-исследовательской деятельности;
- будет сформировано ответственное отношение к учению, готовность и способность обучающихся к саморазвитию и самообразованию, средствами информационных технологий;
- овладеет коммуникативной компетентностью в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности;
- овладеет базой целостного мировоззрения, соответствующего современному уровню развития информационных технологий.

Предметные результаты

По итогам реализации программы, учащиеся будут знать:

- принципы программирования на языке Python;
- терминологию программирования;
- основы дизайнерского оформления созданных программ.
- По итогам реализации программы, учащиеся будут уметь:
- производить чтение и запись программ на языке Python;
- запускать и отлаживать программу.

1.4. Содержание программы
Учебный план дополнительной общеразвивающей программы

№ п/п	Наименование раздела, темы	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
1.	Введение в Python	14	6	8	
1.1.	Язык Python. Среда программирования. Особенности ввода-вывода	4	2	2	Практическое задание, тестирование
1.2.	Типы данных, операции. Оператор присваивания	4	2	2	Практическое задание, тестирование
1.3.	Числа. Стандартные операции	6	2	4	Практическое задание, тестирование
2.	Алгоритмические инструкции	22	6	16	
2.1.	Условный оператор	2	2	-	Практическое задание, тестирование
2.2.	Цикл while. Теория чисел	10	2	8	Практическое задание, тестирование
2.3.	Цикл for	10	2	8	Практическое задание, тестирование
3.	Строки	20	8	12	
3.1.	Литералы строк	4	2	2	Практическое задание, тестирование
3.2.	Срезы строк	4	2	2	Практическое задание, тестирование

3.3.	Методы строк	12	4	8	Практическое задание, тестирование
4.	Функции	20	8	12	
4.1.	Парадигма и преимущества структурного программирования	2	2	-	Практическое задание, тестирование
4.2.	Граф и стек вызовов функций. Области видимости переменных	4	2	2	Практическое задание, тестирование
4.3.	Прямая рекурсия	10	2	8	Практическое задание, тестирование
4.4.	Косвенная рекурсия	4	2	2	Решение задач повышенной трудности
5.	Списки и кортежи	20	8	12	
5.1.	Списки и кортежи в Python. Сходства и различия	2	2	-	Практическое задание, тестирование
5.2.	Операции со списками	10	2	8	Практическое задание, тестирование
5.3.	Срезы списков	4	2	2	Практическое задание, тестирование
5.4.	Матрицы. Операции над матрицами	4	2	2	Практическое задание, тестирование
6.	Словари и множества	8	4	4	Практическое задание, тестирование
6.1.	Словари	4	2	2	Практическое задание, тестирование
6.2.	Множества	4	2	2	Практическое задание, тестирование
7.	Объектно-ориентированное программирование	38	4	34	Практическое задание, тестирование
7.1.	Классы в Python	2	2	-	Практическое задание, тестирование
7.2.	Разработка собственного класса	6	2	4	Практическое задание, тестирование

7.3.	Разработка и программирование собственного проекта	30	-	30	Практическое задание, тестирование
8.	Заключительное занятие	2	-	2	
8.1.	Подведение итогов. Индивидуальный проект.	2		2	Защита проекта
	Итого	144	44	100	

Содержание учебного плана

1. Введение в Python

Основные понятия: трансляция, интерпретация, компиляция, синтаксис, семантика, прагматика, переменная, динамическая типизация, служебные слова, идентификаторы, простые типы данных, приоритеты операций, литералы чисел, операция присваивания, PEP 8.

1.1. Язык Python. Среда программирования. Особенности ввода, вывода

Теория. Язык программирования Python. Достоинства и недостатки.

Области применения. Интерактивный режим работы программы.

Практика. Установка языка программирования Python 3.5 и среды программирования WingIDE 100. Регистрация на Интернет-ресурсах.

1.2. Типы данных, операции. Оператор присваивания

Теория. Ввод и вывод числовой информации.

Практика. Тренировочное задание на ввод и вывод числовой информации.

1.3. Числа. Стандартные операции

Теория. Стандартные операции с целыми и действительными числами.

Стиль программирования Python (PEP 8).

Практика. Решение простых задач в интерактивном режиме.

2. Алгоритмические инструкции

Основные понятия: логический тип данных, логические операции (and, or, not), условный оператор, условное и альтернативное исполнение алгоритма, операторы сравнения, вложенность операторов, оператор цикла, переменная-флаг, генерация псевдослучайной последовательности, инструкции break, continue и pass.

2.1. Условный оператор

Теория. Условная и циклическая инструкции. Каскадность и вложенность алгоритмических инструкций.

Практика. Решение задач на анализ условий

2.2. Цикл while

Теория. Теория чисел. Фильтрация потока чисел. НОД и НОК. Проверка числа на простоту. Алгоритм Евклида. Нахождение максимума и минимума.

Практика. Решение задач на анализ чисел потока и целочисленной арифметики.

2.3. Цикл for

Теория. Теория чисел. Фильтрация потока чисел. НОД и НОК. Проверка числа на простоту. Алгоритм Евклида. Нахождение максимума и минимума.

Практика. Решение задач на анализ чисел потока и целочисленной арифметики повышенной трудности.

3. Строки

Основные понятия: символ, строка, литерал, таблицы кодов ASCII, UTF-8, отладка кода, неизменяемый объект, формат вывода строки, экранированные escape-последовательности, положительная и отрицательная нумерация символов в строке, срез, конкатенация, длина строки.

3.1. Литералы строк

Теория. Понятие «литералы строк».

Практика. Ввод-вывод строки. Решение задач на ввод строки, поиск подстроки.

3.2. Срезы строк

Теория. Форматирование строки.

Практика. Преобразование строки. Решение упражнений.

3.3. Методы строк

Теория. Методы работы со строкой.

Практика. Применение методов строки. Решение задач.

4. Функции

Основные понятия: подпрограмма, функция, процедура, рекурсия, глубина рекурсии, объявление, определение и вызов функции, возврат значений, глобальные и локальные переменные, передача параметров, работа с памятью, граф вызовов, стек вызовов, полиморфизм функций, утиная типизация, lambda-функции.

4.1. Парадигма и преимущества структурного программирования

Теория. Обзор парадигм программирования. Особенности применения языков программирования.

4.2. Граф и стек вызовов функций. Области видимости переменных

Теория. Применение стек и граф вызовов функций.

Практика. Выполнение тренировочных упражнений.

4.3. Прямая рекурсия

Теория. Понятие прямой рекурсии и ее применение.

Практика. Выполнение тренировочных упражнений.

4.4. Косвенная рекурсия

Теория. Понятие косвенной рекурсии и ее применение.

Практика. Выполнение тренировочных упражнений.

Практические занятия к темам 4.2., 4.3., 4.4.: Нахождение суммы чисел. Числа Фибоначчи. Вычисление степени. Ханойские башни. Использование библиотеки математических функций. Решение задач повышенной трудности.

5. Списки и кортежи

Основные понятия: список, кортеж, элемент списка и кортежа, индекс, срез списка, матрица, многомерный список, сортировка, сложность алгоритма, устойчивость сортировки, квадратичная, быстрая, синхронная, поразрядная сортировки списка, случайное перемешивание.

5.1. Списки и кортежи в Python. Сходства и различия

Теория. Представление списка и кортежа в памяти компьютера, сходства и различия.

5.2. Операции со списками

Теория. Способы заполнения списка (с клавиатуры, из файла, случайным образом, по формуле).

Практика. Решение задач на ввод-вывод элементов одномерного и многомерного списка.

5.3. Срезы списков

Теория. Методы работы со списком и кортежем. Методы сортировки списка.

Практика. Решение задач на ввод-вывод элементов одномерного многомерного списка и кортежа, преобразование, поиск, замену, подсчет.

5.4. Матрицы. Операции над матрицами

Теория. Вычисление сложности алгоритма. Многомерные списки.

Практика. Решение задач повышенной трудности.

6. Словари и множества

Основные понятия: *словарь, множество, ключ, кодирование.*

6.1. Словари

Теория. Понятие словаря. Способы создания. Словарь, преимущества и недостатки, методы работы со словарем. Словари со смешанными значениями. Кодирование и декодирование текста.

Практика. Решение задач на заполнение, преобразование, поиск, замену, подсчет, вывод элементов словаря.

6.2. Множества

Теория. Понятие множества. Создание множеств. Множество, преимущества и недостатки, методы работы с множеством.

Практика. Решение задач повышенной трудности.

7. Объектно-ориентированное программирование (ООП)

Основные понятия: ООП, класс, метод INIT, экземпляр, наследование, полиморфизм, исключения, виджет, интерфейс, событие, техническое задание, проект, проектная деятельность, виды проектов.

7.1. Классы в Python

Теория. Понятия «класс», «метод INIT», «экземпляр», «наследование», «полиморфизм», «исключения», «виджет», «интерфейс», «интерфейс», «событие».

7.2. Разработка собственного класса

Теория. Принципы разработки собственного класса. Обработка и генерация исключений. Виджет, методы виджета. Графическая библиотека tkinter, класс Tk. Системные методы. События.

Практика. Создание собственного класса.

7.3. Разработка и программирование собственного проекта

Практика. Выбор вида и темы проекта. Составление технического задания.

Программирование. Разработка технической документации и презентации проекта.

8. Подведение итогов

8.1. Защита индивидуального проекта

Практика. Защита индивидуального проекта.

1.5. Формы аттестации/контроля

Результаты освоения программы отслеживаются по итогам опросов, выполнения практических заданий.

В процессе реализации программы предусмотрены следующие виды контроля:

- **входной контроль проводится** с целью определения уровня знаний учащихся;
- **промежуточный контроль** проводится регулярно на занятиях с целью определения степени усвоения материала в форме опроса, решения задач и практических заданий;
- **итоговый контроль** – защита проекта.

Результаты обучения по программе выявляются по итогам проведения олимпиад, соревнований по программированию, защиты проекта.

2.1. Методическое обеспечение

В программе используются следующие методические материалы: учебно-тематический

план; методическая литература для педагогов дополнительного образования и обучающихся; ресурсы информационных сетей по методике проведения занятий, также реализуется в очной форме с использованием электронных технологий.

Формы организации учебных занятий

- фронтальная работа - одновременная работа со всеми учащимися
- коллективная работа - организация проблемно-поискового или творческого взаимодействия между детьми
- индивидуально-фронтальная работа - чередование индивидуальных и фронтальных форм работы
- групповая работа - организация работы по малым группам (от 2 до 7 человек)
- коллективно-групповая работа - выполнение заданий малыми группами, последующая презентация результатов выполнения заданий и их обобщение
- работа в парах - организация работы по парам
- индивидуальная работа - индивидуальное выполнение заданий, решение проблем

Формы проведения учебных занятий

- беседа;
- демонстрация;
- практикумы начинающего робототехника, включающего проведение лабораторно-практических работ и прикладного программирования

Формы взаимодействия образовательного процесса

- *Диалог* (характеризуется высоким уровнем эмпатии педагога, умением принимать ребенка таким, какой он есть на самом деле, гибкостью мышления педагога, умением адекватно оценивать себя, отсутствием у педагога стереотипов в восприятии обучающихся).
- *Сотрудничество* (предполагает совместное определение целей образования и планирование учебного процесса, совместное распределение сил и средств на основе возможностей педагога и обучающихся, контактность, доброжелательность, мобильность, социальную активность, вежливость, эмоциональную стабильность педагога).
- *Опека*, (включает заботу педагога об обучающихся, социальную активность, высокий уровень эмпатии и умение педагога прийти на помощь).

Приемы, используемые на занятиях. Наглядный. Рассматривание на занятиях готовых примеров программ. Информационно-рецептивный. Совместная деятельность педагога и ребёнка. Репродуктивный. Воспроизводство знаний и способов деятельности (программы по образцу, беседа, упражнения по аналогу). Проблемный. Постановка проблемы и поиск решения. Творческое использование готовых заданий (предметов), самостоятельное их преобразование. Игровой. Использование сюжета игр для организации детской деятельности, персонажей для обыгрывания сюжета. Частично-поисковый. Решение проблемных задач с помощью педагога.

Педагогические технологии:

№	Наименование технологии, методик	Характеристика технологий в рамках образовательной программы
1	Технология группового обучения	С помощью групповой технологии учебная группа, поделённая на подгруппы решает и выполняет конкретные задачи таким образом, что виден вклад каждого обучающегося.
2	Игровая технология	Обеспечивает личностную мотивационную включенность каждого обучающегося, что значительно повышает результативность обучения по программе, т.к. каждый может попробовать себя в разных ролях
3	Здоровьесберегающая технология	Благодаря этим технологиям обучающиеся учатся жить вместе и эффективно взаимодействовать. Они способствуют активному участию самого обучающегося в освоении культуры человеческих

		отношений, в формировании опыта здоровьесбережения, который приобретается через постепенное расширение сферы общения и деятельности ребёнка, становления самосознания и активной жизненной позиции на основе воспитания и самовоспитания, формирования ответственности за свое здоровье, жизнь и здоровье своих товарищей.
4	Электронные (дистанционные) технологии	С помощью этих процессов происходит подготовка и передача информации обучающемуся, через компьютер (дистанционно)

2.2. Условия реализации программы

Материально-техническое обеспечение

Компьютерный класс с доступом в сеть Интернет со следующим оборудованием: компьютерные столы - 12 шт., компьютерные кресла – 12 шт., шкафы встроенные - 1 шт., ноутбуки с программным обеспечением Python, принтер - 1 шт.

Дидактический материал, необходимый для проведения занятий:

краткие конспекты материалов для лекций;
распечатки заданий для практикумов;
презентационные материалы для объяснения;
карточки с индивидуальными заданиями.

Техническое оснащение занятий:

компьютер для демонстрации презентаций;
проектор;
рабочие компьютеры учащихся для работы с доступом в Интернет;
принтер для распечатки заданий.

Кадровое обеспечение: педагог дополнительного образования, имеющий специальную подготовку по профилю программы

2.3. Календарный учебный график
дополнительной общеобразовательной общеразвивающей краткосрочной программы «Программирование на Python»

№ п/п	Месяц	Число	Время проведения	Форма занятия	Кол-во часов	Тема занятия	Место проведения	Форма контроля
1-4				Практикум	4	Язык Python. Среда программирования. Особенности ввода-вывода	Учебный кабинет Компьютерный класс https://telemost.yandex.ru/	Практическое задание, тестирование
5-8				Практикум	4	Типы данных, операции. Оператор присваивания		Практическое задание, тестирование
9-14				Практикум	6	Числа. Стандартные операции		Практическое задание, тестирование
15-16				Практикум	2	Условный оператор		Практическое задание, тестирование
17-26				Практикум	10	Цикл while. Теория чисел		Практическое задание, тестирование
27-36				Практикум	10	Цикл for		Практическое задание, тестирование
37-40				Практикум	4	Литералы строк		Практическое задание, тестирование
41-44				Практикум	4	Срезы строк		Практическое задание, тестирование

45-56				Практикум	12	Методы строк	Учебный кабинет Компьютерный класс https://telemost.yandex.ru/	Практическое задание, тестирование
57-58				Практикум	2	Парадигма и преимущества структурного программирования		Практическое задание, тестирование
59-62				Практикум	4	Граф и стек вызовов функций. Области видимости переменных		Практическое задание, тестирование
63-72				Практикум	10	Прямая рекурсия		Практическое задание, тестирование
73-76				Практикум	4	Косвенная рекурсия		Практическое задание, тестирование
77-78				Практикум	2	Списки и кортежи в Python. Сходства и различия		Практическое задание, тестирование
79-88				Практикум	10	Операции со списками		Практическое задание, тестирование
89-92				Практикум	4	Срезы списков		Практическое задание, тестирование
93-96				Практикум	4	Матрицы. Операции над матрицами		Практическое задание, тестирование
97-100				Практикум	4	Словари		Практическое задание, тестирование
101-104				Практикум	4	Множества		Практическое задание, тестирование
105-106				Практикум	2	Классы в Python		Практическое задание, тестирование

107-112				Практикум	6	Разработка собственного класса
113-142				Практикум	30	Разработка и программирование собственного проекта
143-144				Практикум	2	Подведение итогов. Индивидуальный проект.

Практическое задание, тестирование
Индивидуальный проект
Индивидуальный проект

2.4. Оценочные материалы

Оценочные материалы образовательных результатов

Показатели (оцениваемые параметры)	Степень выраженности оцениваемого качества	Число баллов	Методы диагностики
Теоретические знания по разделам/темам учебно-тематического плана программы	овладел менее чем ½ объема знаний, предусмотренных программой	1	Наблюдение, тестирование, защита работы и др.
	объем усвоенных знаний составляет более ½	2	
	освоил практически весь объем знаний, предусмотренных программой за конкретный период	3	
Практические умения и навыки, предусмотренные программой	овладел менее чем ½ предусмотренных умений и навыков	1	Наблюдение, защита работы
	объем усвоенных умений и навыков составляет более ½	2	
	овладел умениями и навыками, предусмотренными программой за конкретный период	3	

Оценочные материалы личностных результатов

Показатели (оцениваемые параметры)	Степень выраженности оцениваемого качества	Число баллов	Методы диагностики
Сформированность активности, организаторских способностей	мало активен, наблюдает за деятельностью других, забывает выполнить задание. Результативность невысокая	1	Наблюдение
	активен, проявляет стойкий познавательный интерес, трудолюбив, добивается хороших результатов	2	
	активен, проявляет стойкий познавательный интерес, добивается выдающихся результатов, инициативен, организует деятельность других	3	
Сформированность коммуникативных навыков, коллективизма	поддерживает контакты избирательно, чаще работает индивидуально, публично не выступает	1	Наблюдение
	вступает и поддерживает контакты, не вступает в конфликты, дружелюбен со всеми, по инициативе руководителя или группы выступает перед аудиторией	2	
	легко вступает и поддерживает контакты, разрешает конфликты, дружелюбен со всеми, инициативен, по собственному желанию успешно выступает перед аудиторией	3	
Сформированность ответственности, самостоятельности, дисциплинированности	неохотно выполняет поручения. Начинает работу, но часто не доводит ее до конца.	1	Наблюдение

	справляется с поручениями и соблюдает правила поведения только при наличии контроля и требовательности преподавателя; выполняет поручения охотно, ответственно. Хорошо ведет себя независимо от наличия или отсутствия контроля, но не требует этого от других	2	
	выполняет поручения охотно, ответственно, часто по собственному желанию, может привлечь других. Всегда дисциплинирован, везде соблюдает правила поведения, требует того же от других	3	
Сформированность креативности, склонности к самостоятельному творчеству	может работать в проектно-исследовательской группе при постоянной поддержке и контроле. Способен принимать творческие решения, но в основном использует традиционные способы	1	Наблюдение
	может разработать свой творческий проект с помощью педагога. Способен на творческие решения, но в основном использует традиционные способы	2	
	высокий творческий потенциал. Самостоятельно выполняет работы. Находит нестандартные решения, новые способы выполнения заданий	3	

Оценочные материалы метапредметных результатов

Показатели (оцениваемые параметры)	Степень выраженности оцениваемого качества	Число баллов	Методы диагностики
Понимать и принимать учебную задачу, сформулированную педагогом	овладел менее чем 1/2 объема задач, предусмотренных программой	1	Наблюдение
	объем усвоенных задач составляет более 1/2	2	
	демонстрирует полное понимание, предусмотренных программой задач за конкретный период	3	
Планировать свои действия на отдельных этапах работы над выполнением творческого задания	овладел менее чем 1/2 объема знаний, предусмотренных программой	1	Наблюдение
	демонстрирует неполное освоение планируемых действий, но более 1/2	2	

	освоил план действий в заданных условиях	3	
Осуществлять контроль, коррекцию и оценку результатов своей деятельности; понимать и применять полученную информацию при выполнении заданий	знает, но избегает их употреблять в деятельности	1	Наблюдение
	демонстрирует неполное освоение заданных параметров, но более ½	2	
	освоил план действий в заданных условиях	3	

Мониторинг результатов обучения ребенка по дополнительной общеразвивающей программе

Показатели (оцениваемые параметры)	Методы диагностики
1. Уровни знаний / пониманий <ul style="list-style-type: none"> ▪ Наличие общих представлений (менее ½ объема знаний) ▪ Наличие ключевых понятий (объем усвоенных знаний более 1/2) ▪ Наличие прочных системных знаний, (освоен практически весь объем) 	Наблюдение, тестирование, контрольный опрос, собеседование
2. Уровни умения применять знания на практике <ul style="list-style-type: none"> ▪ Репродуктивный несамостоятельный (деятельность осуществляется под непосредственным контролем преподавателя на основе устных и письменных инструкций). ▪ Репродуктивный самостоятельный (деятельность осуществляется на основе типовых алгоритмов). ▪ Творческий (в процессе деятельности творчески используются знания, умений, предлагаются и реализуются оригинальные решения) 	Контрольное задание
3. Наличие опыта самостоятельной деятельности <ul style="list-style-type: none"> ▪ Очень незначительный опыт; ▪ Незначительный балл (от случая к случаю); ▪ Эпизодическая деятельность; ▪ Периодическая деятельность; ▪ Богатый опыт (систематическая деятельность) 	Анализ, исследовательские работы, конкурсные работы, наблюдение
4. Сформированность личностных качеств <ul style="list-style-type: none"> ▪ Очень низкая (проявились отдельные элементы); ▪ Низкая (проявилась частично); ▪ Недостаточно высокая (проявилась в основном); ▪ Высокая (проявились полностью) 	Анализ, наблюдение, собеседование

На основе вышеприведенного анализа заполняется диагностическая карта (оценочный лист) таблица 2.

Диагностическая карта успеваемости воспитанников объединения

Ф.И.О.	Знать / понимать (макс-3 балла)					Уметь использовать (макс-4 балла)					Владеть опытом (макс-5 баллов)					Личностные качества (макс-4 балла)					Итого баллов	Оценка
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5		
Иванов А.																						

Результаты деятельности каждого обучающегося по каждому из показателей суммируются для определения итогового балла. Показатель усвоения (продуктивности обучения) вычисляется по формуле:

$$K_{\text{усв}} = \Phi/\Pi * 100\%$$

Где К усв- коэффициент усвоения
Ф – фактический объем знаний (набранная сумма баллов)
П – полный объем знаний (максимальная сумма баллов).
В дальнейшем можно перейти к пятибалльной системе оценки.

Коэффициент сформированности:
80-100 - «отлично»
50-79 - «хорошо»
30-49 - «удовлетворительно»
Менее 29 - «неудовлетворительно»

2.5. Список литературы

Для педагога

1. Бэрри П. Изучаем программирование на Python. — М., 2017. — 624 с.
2. Буйначев С. К. Основы программирования на языке Python: учебное пособие. — Екатеринбург: Изд-во Урал. ун-та, 2014. — 91 с.
3. Бхаргава А. Грожаем алгоритмы: иллюстрированное пособие для программистов и любопытствующих. — СПб.: Питер, 2017. — 288 с.
4. Гэддис Т. Начинаем программировать на Python / пер. с англ. 4-е изд. — СПб.: БХВ-Петербург, 2019. — 768 с.
5. Доусен М. Программируем на Python / М. Доусен - СПб.: Питер, 2016. -416с.
6. Мюллер Дж. Python для чайников. — СПб. : Диалектика, 2019. — 416с.
7. Луридад П. Алгоритмы для начинающих: теория и практика для разработчика. — М. : Эксмо, 2018. — 608 с.
8. Лутц М. Изучаем Python, пер. с англ. 3-е изд. — СПб.: Символ Плюс, 2009. — 848 с.
9. Любанович Б. Простой Python. Современный стиль программирования / Б. Любанович. - СПб.: Питер, 2016. - 480с.
10. Прохоренко Н.А., Дронов В.А. Python 3 и PyQt 5. Разработка приложений / Н.А. Прохоренко, В.А. Дронов - СПб.: «БХВ- Петербург», 2016. - 832с.
11. Рафгарден Т. Совершенный алгоритм. Жадные алгоритмы и динамическое программирование. — СПб.: Питер, 2020. — 256 с.
12. Рейтц К., Шлюссер Т. Автостопом по Python. — СПб. : Питер, 2017. — 336 с.
13. Саммерфильд М. Python на практике / М. Саммерфильд, пер. А.А. Слинкин – М.: ДМК-Пресс, 2014. - 338с.
14. Фёдоров Д. Ю. Программирование на языке высокого уровня Python: учебное пособие для прикладного бакалавриата. — М. : Издательство Юрайт, 2019. — 161 с.

Для обучающихся

1. Python 3 для начинающих: <https://pythonworld.ru/samouchitel-python>
2. Учебник по языку программирования Python (хабраиндекс): <https://habr.com/ru/post/61905/>
3. Python/Учебник Python
https://ru.wikibooks.org/wiki/Python/%D0%A3%D1%87%D0%B5%D0%B1%D0%BD%D0%B8%D0%BA_Python_3.1
4. Python для начинающих 2021 — уроки, задачи и тесты:
<https://pythonru.com/uroki/python-dlja-nachinajushhih>